21世纪高等学校计算机类课程创新规划教材 · 微课版



MySQL数据库应用与开发

习题解答与上机指导

◎ 姜桂洪 主编 孙福振 苏晶 编著

清华大学出版社

21世纪高等学校计算机类课程创新规划教材·微课版

MySQL数据库应用与开发 习题解答与上机指导

姜桂洪 主编
孙福振 苏 晶 编著

清华大学出版社
北 京

内 容 简 介

本书是《MySQL 数据库应用与开发》的配套教材。全书包括 3 部分内容：第 1 部分是《MySQL 数据库应用与开发》的习题参考答案以及本部分内容的教学重点和难点简介；第 2 部分是 MySQL 系统的常用实践环境及可视化软件的详细介绍和实验指导，提供了 16 个有着详细操作步骤的实验；第 3 部分是 6 套涵盖所有 MySQL 内容的模拟试题及参考答案。

本书体系完整、结构安排合理、内容翔实，注重实践与理论相结合，操作步骤讲解清楚，实验步骤规范且内容有代表性，并注意在实践过程中引导学生进行自我创新学习。

本书适合作为高等院校本科、专科计算机及相关专业的学生学习数据库应用系统开发技术的辅导教材，也可作为从事数据库管理与开发的科技工作者、参加 MySQL 等级考试的读者的参考用书。

图书在版编目(CIP)数据

MySQL 数据库应用与开发习题解答与上机指导/姜桂洪主编. —北京：清华大学出版社，2018
(21 世纪高等学校计算机类课程创新规划教材·微课版)
ISBN 978-7-302-49586-4

Ⅰ. ①M…　Ⅱ. ①姜…　Ⅲ. ①SQL 语言－高等学校－教学参考资料　Ⅳ. ①TP311.132.3

中国版本图书馆 CIP 数据核字(2018)第 027463 号

责任编辑：魏江江　薛　阳
封面设计：刘　键
责任校对：时翠兰
责任印制：丛怀宇

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：14.25　　**字　　数**：338 千字
版　　次：2018 年 8 月第 1 版　　**印　　次**：2018 年 8 月第1 次印刷
印　　数：1～1500
定　　价：39.00 元

产品编号：076086-01

前　言

本书是《MySQL 数据库应用与开发》(姜桂洪主编,2018 年由清华大学出版社出版)的配套辅导教材。书中采用目前实际生产和教学领域内应用广泛的 MySQL 软件,利用内容丰富的习题答案、涵盖 MySQL 所有重要知识点的实验基本操作、典型的模拟试题等形式,从多视角为初学者提供了在 Windows 系列平台上学习数据库知识的服务环境。同时,还通过可视化软件及综合性实验介绍了 MySQL 与 PHP 平台相结合实现数据管理的常用操作。

本书从基本操作着手,从教学实际需求出发,结合初学者的认知规律,由浅入深、精心选择 MySQL 数据库的知识点,以数据库及相关数据库对象为基础,以备份、还原、日志管理、性能优化和利用 PHP 语言访问 MySQL 数据等数据库系统管理组件为目标,以形式多样、内容丰富的习题和参考答案对基本操作进行示范,与主教材一起打造出比较完整的立体化课程体系。

本书在模块内容安排上体现循序渐进、重点突出的特点。习题解答部分中,每章都配有教学内容的重点、难点介绍和详细的习题参考答案,主要目的是帮助学生掌握数据库的基本概念和 MySQL 语句的基本规则和基本操作。实验部分主要帮助学生进一步强化基本单元操作,培养学生独立操作的能力。模拟题可以让学生在掌握好所学知识的基础上,进一步了解自己学习本课程的知识水平。通过本书提供的指导材料,学生可以在自由上机的环境下,完成相关数据库的常用操作及管理。

全书总共包括如下 3 部分内容:

第 1 部分包括第 1～15 章,内容为主教材的习题参考答案,所有编程题的答案均全部通过调试。

第 2 部分包括第 16～18 章,内容为 MySQL 软件的常用实践环境配置操作、MySQL 的可视化软件 Navicat for MySQL 和 phpMyAdmin 的详细介绍和基本操作,以及学生上机实验时需要注意的问题,并根据教材内容提供 16 个配套实验。实验操作步骤清晰、完整且有代表性。

第 3 部分提供 6 套模拟试题及参考答案,为读者学习 MySQL 数据库提供了自我检查知识掌握水平的契机,旨在帮助读者了解和检验自己的学习情况。

全书由姜桂洪、孙福振、苏晶等老师参与编写,由姜桂洪统稿。在本书编写过程中还参阅了大量的数据库方面的文献和网站资料,在此对提供者一并深表感谢。

由于作者水平有限,书中纰漏之处在所难免,恳请读者批评指正。

编　者

2018 年 1 月



第 1 部分 《MySQL 数据库应用与开发》习题参考答案

第1部分
《MySQL数据库应用与开发》
习题参考答案

学习导读：

本部分包括《MySQL数据库应用与开发》第1～15章的教学重点和难点提示，以及所有习题的参考答案，所有代码均在MySQL 5.7.17版本的环境下测试通过。

每章习题包括选择题、思考题和上机练习题3种类型：

（1）选择填空注重基本概念和基本操作的知识点掌握。

（2）思考题侧重基本概念的理解。

（3）上机练习题侧重的是基本操作的练习。第4章的习题答案还附加了主教材和指导书常用数据库teaching的表结构和数据，供大家在学习过程中参考。

建议在学习过程中，通过习题熟悉MySQL数据库的基本概念和基本操作。不要满足于看懂程序或习题答案，要多动手做练习，并且要达到一定的做题速度和正确率。在此基础上，借助于网络或其他工具，举一反三，解决一些比教材习题更新颖、算法更复杂的问题。

第 1 章　MySQL 数据库概述

1.1　教学要求

1.1.1　基本要求

（1）掌握有关数据库的基本概念及其之间的关系，如数据、信息、数据库、数据库管理系统、数据库系统和 SQL 语言。

（2）掌握有关数据库的基本关系理论，投影、选择和连接的内涵。

（3）概念模型的实体、属性、联系、主键、实体集、实体型的基本概念和含义，了解概念模型的表示方法。了解数据库设计中关系模型的规范化的意义。

（4）了解 MySQL 数据库软件的特点和用途。

（5）熟练掌握 MySQL 数据库软件的下载、安装和配置过程。

（6）学会利用客户端工具登录 MySQL 数据库。

（7）了解可视化工具 MySQL Workbench 的安装和使用方法。

（8）积累本课程的常用英文词汇。

1.1.2　重点与难点

1. 本章重点

（1）数据库管理系统的功能和组成。

（2）关系数据库的基本理论。

（3）数据库系统的基本组成。

（4）安装、启动和配置 MySQL 的基本过程。

（5）利用客户端工具登录 MySQL 数据库。

2. 本章难点

（1）MySQL 数据库软件的下载、安装和配置过程。

（2）利用客户端工具登录 MySQL 数据库。

（3）关系模型的规范化设计的用途。

（4）概念模型的表示方法 E-R 图的实现。

1.2　习题参考答案

1. 选择题

（1）数据模型的 3 要素不包括________。

A. 数据结构　　B. 数据操作　　C. 数据类型　　D. 完整性约束

(2) 关系运算不包括________。

A. 连接　　B. 投影　　C. 选择　　D. 查询

(3) 主教材表 1-1 所示的学生信息表中的主键为________。

A. studentno　　B. sex　　C. birthdate　　D. sname

(4) 下面的数据库产品中,________是开源数据库。

A. Oracle　　B. SQL Server　　C. MySQL　　D. DB2

(5) E-R 概念模型中,信息的三种概念单元不包括________。

A. 实体型　　B. 实体值　　C. 实体属性　　D. 实体间联系

(6) Linux 和 UNIX 操作平台上不能使用________作为数据库。

A. Oracle　　B. DB2　　C. MySQL　　D. SQL Server

(7) E-R 图是数据库设计的工具之一,一般适用于建立数据库的________。

A. 概念模型　　B. 结构模型　　C. 物理模型　　D. 逻辑模型

(8) SQL 语言又称________。

A. 结构化定义语言　　B. 结构化控制语言

C. 结构化查询语言　　D. 结构化操纵语言

(9) 从 E-R 模型向关系模型转换,一个 $M:N$ 的联系转换成一个关系模式时,该关系模式的键是________。

A. M 端实体的键　　B. N 端实体的键

C. M 端实体键与 N 端实体键的组合　　D. 重新选取其他属性

(10) DB、DBS 和 DBMS 3 者之间的关系是________。

A. DB 包括 DBMS 和 DBS　　B. DBS 包括 DB 和 DBMS

C. DBMS 包括 DB 和 DBS　　D. 不能相互包括

本题答案:C D A C B D A C C B

2. 简答题

(1) 什么是数据库管理系统?举出日常生活中一些应用数据库的实际范例。

答:数据库管理系统 DBMS 是位于用户和操作系统之间的一种操纵和管理数据库的软件,用于建立、使用和维护数据库。它对数据库进行统一的管理和控制,以保证数据库的安全性和完整性,是数据库系统的核心。

日常生活中,像学生信息管理、网上购物、银行存取款、购买火车票等,其软件的后台都是由数据库作为支撑的。

(2) 说明 MySQL 数据库管理系统基本系统架构拥有哪 4 大模块。

答:数据库管理系统是由多种不同的程序模块组成的,基本数据库管理系统的系统架构一般包括如下 4 部分功能模块:

① 存储管理(Storage Manager)。数据库管理系统通常会自行配置磁盘空间,将数据存入存储装置的数据库。

② 查询处理(Query Processor)。负责处理用户下达的查询语言命令语句,可以再细分

成多个模块负责检查语法、优化查询命令的处理程序。

③ 事务管理(Transaction Manager)。事务管理主要分为：同名的事务管理子系统，负责处理数据库的事务，保障数据库商业事务的操作需要一并执行；锁定管理(Lock Manager)，也称为并发控制管理(Concurrency-Control Manager)子系统，负责资源锁定。

④ 恢复管理(Recovery Manager)。恢复管理主要是日志管理(Log Manager)子系统，负责记录数据库的所有操作，包含事务日志，以便同名的恢复管理子系统能够执行恢复处理，恢复数据库系统存储的数据到指定的时间点。

(3) 举例说明 3 种关系运算的特点。

答：在关系中访问所需的数据时，需要对关系进行一定的关系运算。关系数据库主要支持选择、投影和连接关系运算，它们源于关系代数中并、交、差、选择、投影和连接等运算。

① 选择关系用于查找符合条件的行，从而形成新关系。新关系和原来关系结构一致，记录行变少。例如在学生表中查询年龄 18 岁以上的学生记录。

② 投影关系是从一个表中找出若干字段形成一个新关系的操作，新关系与原来关系的记录行应该一样多，但其数据列变少。例如在学生表中输出学生的学号、姓名和出生日期。

③ 连接关系是从两个表中选择满足条件的记录，且新关系中包含原来多个关系的列，新关系结构中的列类型由原来关系中的列映射而来。例如，查询学生的学号、姓名和期末成绩。3 个输出项并不是在一个表中可以搜索到，需要进行连接操作。即将 2 个或多个表中的行按一定的条件横向结合，形成一个新的临时表，在此临时表中进行数据查询。

3. 上机练习

(1) 从 MySQL 的官方网站(https://dev.mysql.com/downloads/)下载 MySQL 5.7.19 版本的数据库，解压缩后进行安装，观察安装成功后的效果。

答：① 下载软件。登录 MySQL 的官方网站(https://dev.mysql.com/downloads/)，在该网站上访问如下页面：https://dev.mysql.com/downloads/windows/installer/8.0.html。即可得到图 1-1 所示的下载软件版本 mysql-installer-community-5.7.19.0.msi。单击 Download 按钮，即可下载 MySQL 免费软件。

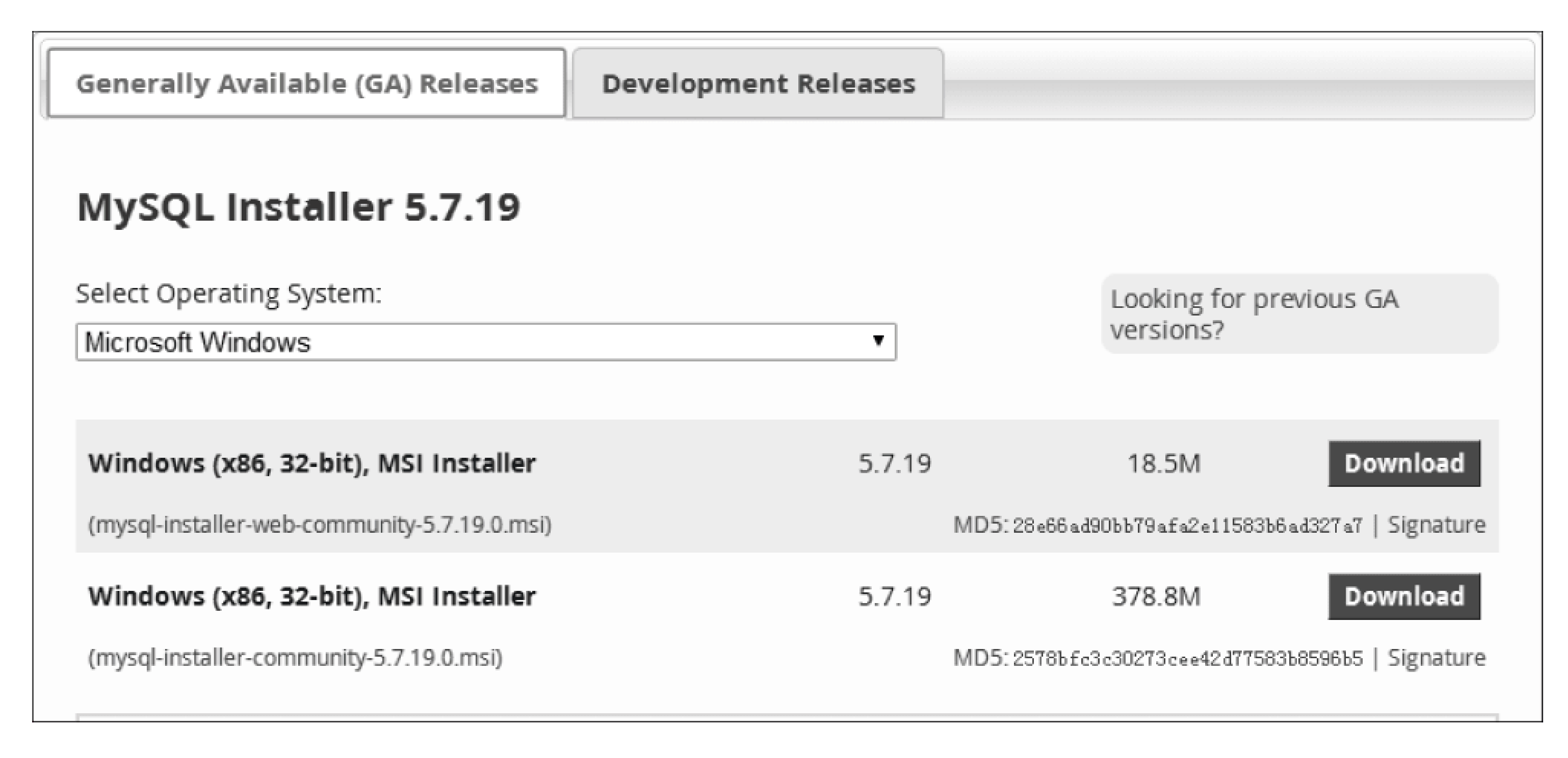


图 1-1 MySQL 5.7.19 版本软件

② 软件安装演示。双击下载软件 mysql-installer-community-5.7.19.0.msi，进入如图 1-2 所示的开始安装界面。单击“运行”按钮，进入许可协议界面 License Agreement，选择 I accept the licence terms 复选框，单击 Next 按钮，进入如图 1-3 所示的选择安装类型界面 Choosing a Setup Type，选择安装类型 Full，单击 Next 按钮，进入图 1-4 所示的界面。单击 Execute 按钮，依次进行安装即可。在实际安装过程中，要注意的问题包括软件环境是否合适，还要注意记住设置参数，如密码、端口号等。



图 1-2 安全警告

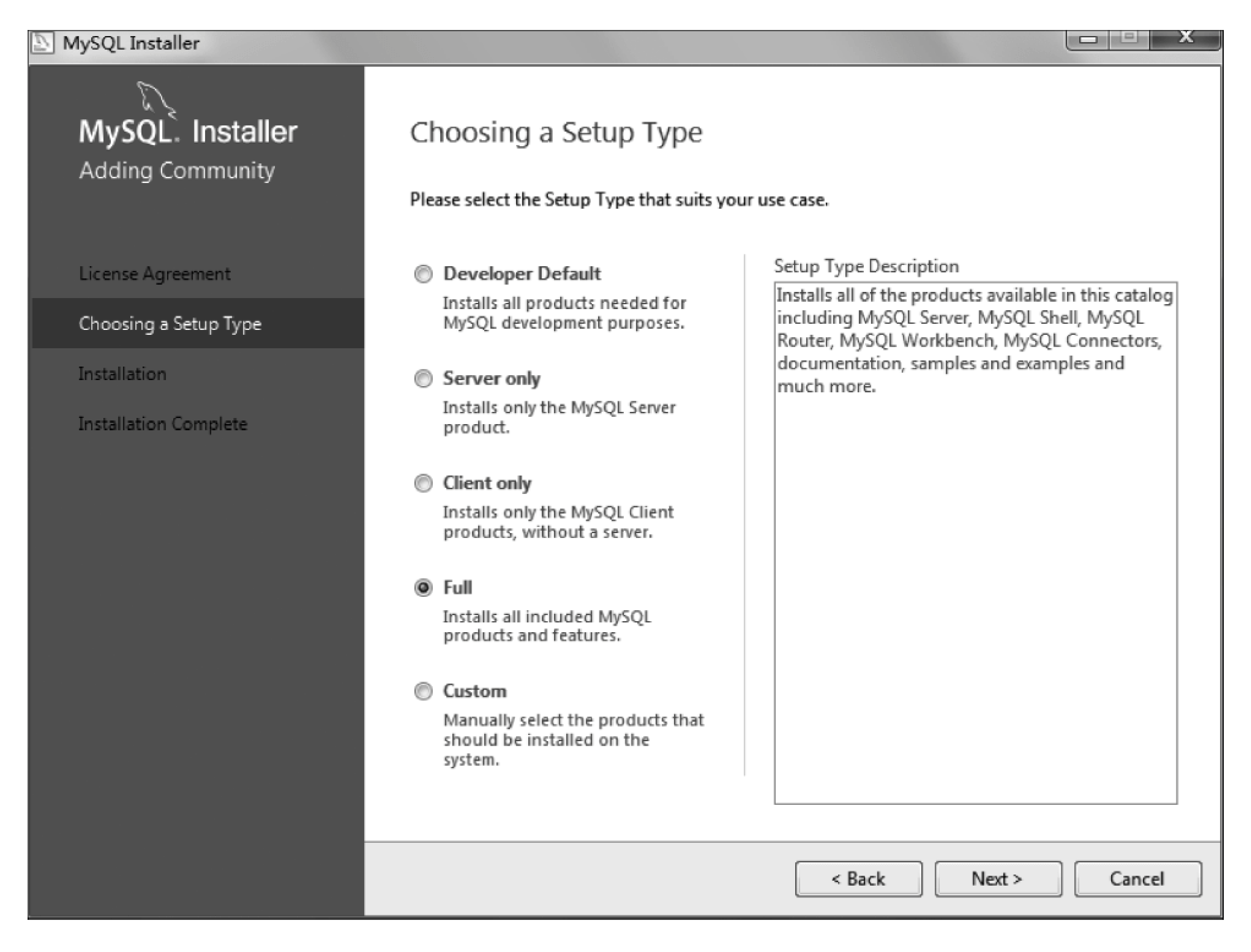


图 1-3 选择安装版本

③ 软件配置演示。右击“计算机”图标，在弹出的快捷菜单中选择“属性”命令，在弹出的“控制面板”主页中单击“高级系统设置”标签，对话框中选择“高级系统设置”，弹出“系统属性”对话框，如图 1-5 所示。

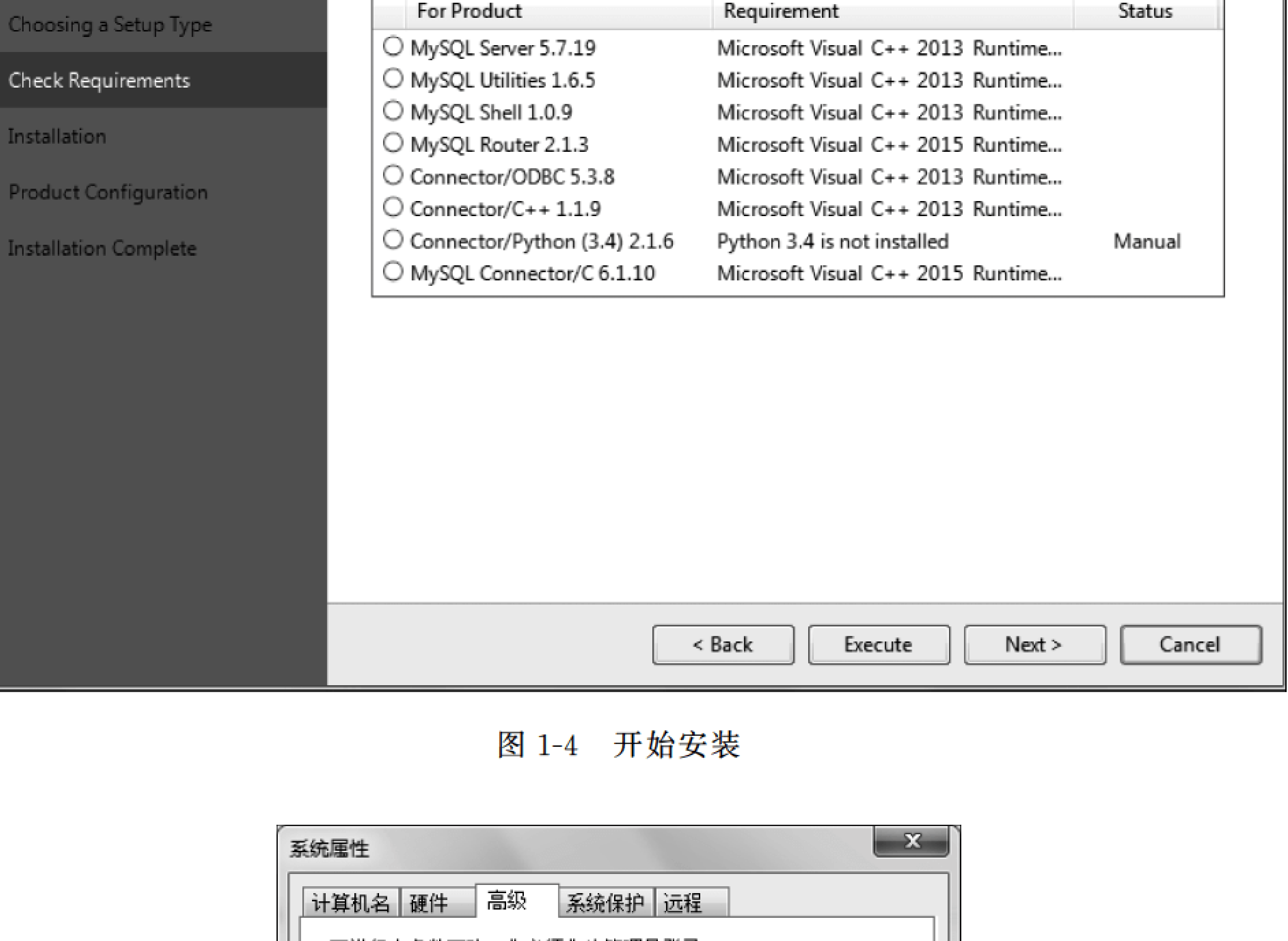


图 1-4　开始安装

系统属性
计算机名 硬件 高级 系统保护 远程
要进行大多数更改，您必须作为管理员登录。
性能
视觉效果，处理器计划，内存使用，以及虚拟内存
设置(S)...
用户配置文件
与您登录有关的桌面设置
设置(E)...
启动和故障恢复
系统启动、系统失败和调试信息
设置(T)...
环境变量(N)...
确定 取消 应用(A)

图 1-5　“系统属性”对话框

④ 在“系统属性”对话框中，打开“高级”选项卡，单击“环境变量”按钮，弹出“环境变量”对话框。在“环境变量”对话框中，定位到“系统变量”中的 path 选项，单击“编辑”按钮，将弹出“编辑系统变量”对话框，如图 1-6 所示。在“编辑系统变量”对话框中，将 MySQL 服务器的 bin 文件夹位置(F:\Program　Files\MySQL\MySQL Server 5.7\bin)添加到变量值文

本框中,注意要使用“;”与其他变量值进行分隔,最后,单击“确定”按钮。环境变量设置完成后,再使用MySQL命令即可成功连接MySQL服务器。



图1-6 “编辑系统变量”对话框

(2) 在控制台中完成操作。在启动MySQL服务后打开控制台,在控制台中输入密码完成MySQL的登录,并且执行select命令的相关语句查看系统的当前日期。

答:主要操作命令和运行结果如下:

```
C:\Users\Administrator> mysql -u root -p
Enter password: ******
mysql> select curdate();
          +------------+
          | curdate()  |
          +------------+
          | 2017-05-01 |
          +------------+
          1 row in set (0.07 sec)
```

(3) 说明在“服务”窗口中,MySQL服务器的启动和关闭方法。

答:执行“开始”→“运行”命令,输入“services.msc”后按Enter键,也可以弹出图1-7所示的窗口。从中可以看到MySQL服务已经启动,而且服务的启动类型是自动启动。



图1-7 “服务”窗口

选中MySQL服务项右击,弹出如图1-8所示的对话框。可以在弹出的菜单中,对MySQL57服务器进行“启动”“停止”“暂停”“恢复”和“重新启动”等操作。



图 1-8　更改服务的启动类型

第 2 章　MySQL 语言基础

2.1　教 学 要 求

2.1.1　基本要求

(1) 了解字符集和字符序的基本概念和用途。

(2) 掌握标识符、关键字、常量、变量等 MySQL 的基本语法要素的概念和应用。

(3) 熟练掌握常用字符型、数值型、日期时间型等数据类型的用途。

(4) 熟练掌握算术运算符、比较运算符、逻辑运算符及位运算符的用途。了解运算符的优先级及其作用。

(5) 熟练掌握常用字符型函数、日期时间型函数、数值型函数的格式和用途。

(6) 掌握各种值类型表达式的写法和用途。

(7) 学会利用 select 语句对常量、变量、函数、表达式等进行数据输出的操作。

2.1.2　重点与难点

1. 本章重点

(1) 字符集和字符序的基本概念。

(2) 标识符、关键字、常量、变量等 MySQL 的基本语法要素的概念和应用。

(3) 常用字符型、数值型、日期时间型等数据类型的用途。

(4) 算术运算符、比较运算符、逻辑运算符及位运算符的用途。

(5) 常用字符型函数、日期时间型函数、数值型函数的格式和用途。

2. 本章难点

(1) 各种数据类型的变量的定义和使用方法。

(2) 常用字符型函数、日期时间型函数、数值型函数的格式。

(3) 算术运算符、比较运算符、逻辑运算符的使用方法。

2.2　习题参考答案

1. 选择题

(1) 以下命令中，________是 DML 语句。

A. create　　　B. alter　　　C. select　　　D. drop

(2) 以下关于 MySQL 的说法中错误的是________。

A. MySQL 是一种关系型数据库管理系统

B. MySQL 软件是一种开放源码软件

C. MySQL 服务器工作在客户端/服务器模式下,或嵌入式系统中

D. 在 Windows 系统下 MySQL 中书写 MySQL 语句区分大小写

(3) 控制台中执行________语句时可以退出 MySQL。

A. exlt　　B. go 或 quit　　C. go 或 exit　　D. exit 或 quit

(4) 关于 MySQL 数据库的说法,选项________的说法是错误的。

A. MySQL 数据库不仅开放源码,而且能够跨平台使用。例如,可以在 Windows 操作系统中安装 MySQL 数据库,也可以在 Linux 操作系统中使用 MySQL 数据库

B. MySQL 数据库启动服务时有两种方式,如果服务已经启动可以在任务管理器中查找 mysqlld.exe 程序,如果该进程存在则表示正在运行

C. 手动更改 MySQL 的配置文件 my.ini 时,只能更改与客户端有关的配置,而不能更改与服务器端相关的配置信息

D. 登录 MySQL 数据库成功后,直接输入"help;"语句后,按 Enter 键可以查看帮助信息

(5) 下列________类型不是 MySQL 中常用的数据类型。

A. int　　B. var　　C. time　　D. char

(6) 在 MySQL 中会话变量前面的字符为________。

A. 空格　　B. #　　C. @@　　D. @

(7) 设置表的默认字符集关键字是________。

A. default　character　　B. default　set

C. default　　D. default character set

本题答案:C D D C B D D

2. 简答题

(1) 简述字符集 UTF-8 和 GB 2312 的区别。

答:UTF-8 被称为通用转换格式,是针对 Unicode 字符的一种变长字符编码。该字符集是用以解决国际上字符的一种多字节编码,它对英文使用 8B,中文使用 24B 来编码。UTF-8 包含全世界所有国家需要用到的字符,是国际编码,通用性强。UTF-8 编码的文字可以在各国支持 UTF-8 字符集的浏览器上显示。

GB 2312 是简体中文字符集,GBK 是对 GB 2312 的扩展,GBK 是在国家标准 GB 2312 基础上扩容后兼容 GB 2312 的标准。GBK 的文字编码是用双字节来表示的,即不论中英文字符均使用双字节来表示,为了区分中文,将其最高位都设定成 1。GBK 包含全部中文字符,是国家编码,通用性比 UTF-8 差,不过 UTF-8 占用的数据库比 GBK 大。

(2) datetime 类型和 timestamp 类型的相同点和不同点是什么?

答:日期与时间类型是为了方便在数据库中存储日期和时间而设计的,datetime 和

timestamp 表示日期和时间。其中的每种类型都有其取值的范围，如赋予它一个不合法的值，将会被“0”代替。datetime 的取值范围为 1000-01-01 00:00:00 ～9999-12-31 23:59:59，而 timestamp 的取值范围是 1970-01-01 00:00:00 至 2037 年的某个时间。

(3) MySQL 支持的数据类型主要有哪几类？17 属于什么类型？'17'属于什么类型？

答：MySQL 支持的数据类型主要包括数值型、字符型和日期时间型等。17 属于数值型中的整型数据，'17'属于字符型数据。

(4) 简述系统变量、全局变量和会话变量的关系。

答：系统变量包括全局系统变量和会话系统变量两种类型。

全局变量在 MySQL 启动时由服务器自动将它们初始化为默认值，主要影响整个 MySQL 实例的全局设置，大部分全局变量都是作为 MySQL 的服务器调节参数存在。对全局变量的修改会影响到整个服务器。

会话变量在每次建立一个新的连接时，由 MySQL 来初始化。对会话变量的修改，只会影响到当前的会话，也就是当前的数据库连接。会话变量可以随时定义，必须在名称前加@符号。

(5) 简述聚合函数的特点和用途。

答：聚合函数用于对一组值进行计算并返回一个单一的值。除 count 函数之外，其他聚合函数忽略空值。聚合函数经常与 select 语句的 group by 子句一同使用。聚合函数的作用是在结果集中通过对被选列值的收集处理，并返回一个数值型的计算结果。

3. 上机练习

(1) 创建文本文件。将查询系统当前日期、当前时间以及到 2019 年 1 月 1 号还有多少天。然后通过 MySQL 命令执行文本文件中的内容。

答：在文本文件“d:\ my231. txt”中的内容如下：

```
select now();                          -- 查询当前日期命令
select curtime();                      -- 输出当前时间
select datediff('2019 - 1 - 1',now()); -- 到 2019 年 1 月 1 号的天数
```

在 DOS 窗口中执行 MySQL 命令语句，指定文本文件并执行该文件中的语句。

输出结果如下：

```
C:\Users\Administrator> mysql < D:\my231.txt  - u root  - p
Enter password: ******
now()
2017 - 08 - 22 09:03:53
curtime()
09:03:53
datediff('2019 - 1 - 1',now())
497
C:\Users\Administrator>
```

(2) 利用随机函数输出 20～90 的任意两个数(含 2 位小数)。

答：命令和运行结果如下：

```
mysql> select  20 +  round(70 * rand(),2) , 20 +  round(70 * rand(),2);
       +---------------------+---------------------+
```

```
| 20 + round(70 * rand(),2)    |  20 + round(70 * rand(),2) |
+----------------------+---------------------+
|                47.69       |                20.68       |
+----------------------+---------------------+
1 row in set (0.02 sec)
```

(3) 计算 1000 天后的日期和 3000 分钟后的日期时间。

答：命令和运行结果如下：

```
mysql > select date_add(now(),interval 1000 day);
        +---------------------------+
        | date_add(now(),interval 1000 day)  |
        +---------------------------+
        | 2020 - 03 - 29 19:29:22          |
        +---------------------------+
        1 row in set (0.00 sec)

mysql > select date_add(now(),interval 3000 minute);
        +-----------------------------+
        | date_add(now(),interval 3000 minute)   |
        +-----------------------------+
        | 2017 - 07 - 05 21:29:22               |
        +-----------------------------+
        1 row in set (0.00 sec)
```

第3章 MySQL 数据库的基本操作

3.1 教学要求

3.1.1 基本要求

(1) 了解数据库设计的基本过程。
(2) 掌握概念模型的设计过程和利用 E-R 图描述概念模型的方法。
(3) 了解常用 MySQL 数据库的用途。
(4) 了解常用 MySQL 数据库存储引擎的特点。
(5) 掌握设置和修改 MySQL 数据库存储引擎的操作。
(6) 掌握 MySQL 数据库的创建、打开和存储方法。
(7) 学会利用 MySQL Workbench 可视化软件维护数据库。

3.1.2 重点与难点

1. 本章重点

(1) 概念模型的设计过程。
(2) 利用 E-R 图描述概念模型的方法。
(3) 设置和修改 MySQL 数据库存储引擎的操作。
(4) 掌握 MySQL 数据库的创建、打开、删除和存储方法。
(5) 利用 MySQL Workbench 可视化软件维护数据库的基本操作。

2. 本章难点

(1) 概念模型的设计过程。
(2) 利用 E-R 图描述概念模型的方法。
(3) 利用 MySQL Workbench 可视化软件维护数据库的基本操作。

3.2 习题参考答案

1. 选择题

(1) 在 MySQL 数据库中,通常使用________语句来指定一个已有数据库作为当前工作数据库。

A. using　　B. used　　C. uses　　D. use

(2) ________命令用于删除一个数据库。

A. create database　　　　B. drop database

C. alter database　　　　D. use InnoDB

(3) ________存储引擎支持外键(Foreign Key)和事务(Transaction)和全文检索。

A. MEMORY　　B. MyISAM　　C. InnoDB　　D. MySQL

(4) 在创建数据库时,可以使用________子句确保如果数据库不存在就创建它,如果存在就直接使用它。

A. if not exists　　B. if exists　　C. if not exist　　D. if　exist

(5) 在图形管理工具 MySQL WorkBench 窗口中使用可视化的界面不能够支持________。

A. 数据库引擎开发　　B. 创建数据库　　C. 维护数据库　　D. 数据库设计

(6) MySQL 自带数据库中,________存储了系统的权限信息。

A. information_schema　　　　B. mysql

C. sakila　　　　D. performance_schema

本题答案:D B C A A B

2. 简答题

(1) 如何在 MySQL Workbench 窗口中修改数据库的字符集?

答:连接 MySQL Workbench 软件后,在 SCHEMAS 区选择数据库,例如 mysqltest,右击 mysqltest 数据库,如图 3-1 所示,执行 Alter Schema 命令,进入如图 3-2 所示的数据库修改界面,选择 Collation 字符集选项即可。

(2) 简述在 MySQL Workbench 窗口中创建数据库的步骤。

答:MySQL Workbench 窗口中使用可视化的界面可以创建数据库。使用 MySQL Workbench 创建数据库的步骤如下:

① 进入 MySQL Workbench 主界面进行 MySQL 数据库的连接。进入 MySQL Workbench 数据库操作的主界面,单击如图 3-2 所示的提示创建数据库命令。

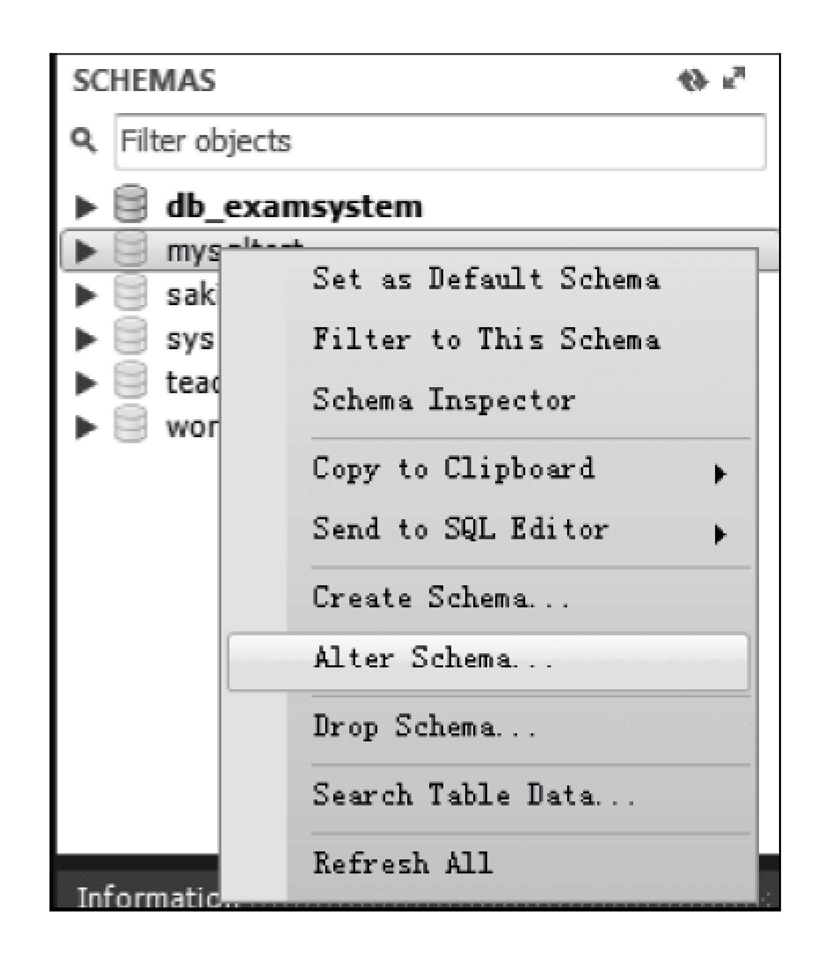


图 3-1　选择数据库

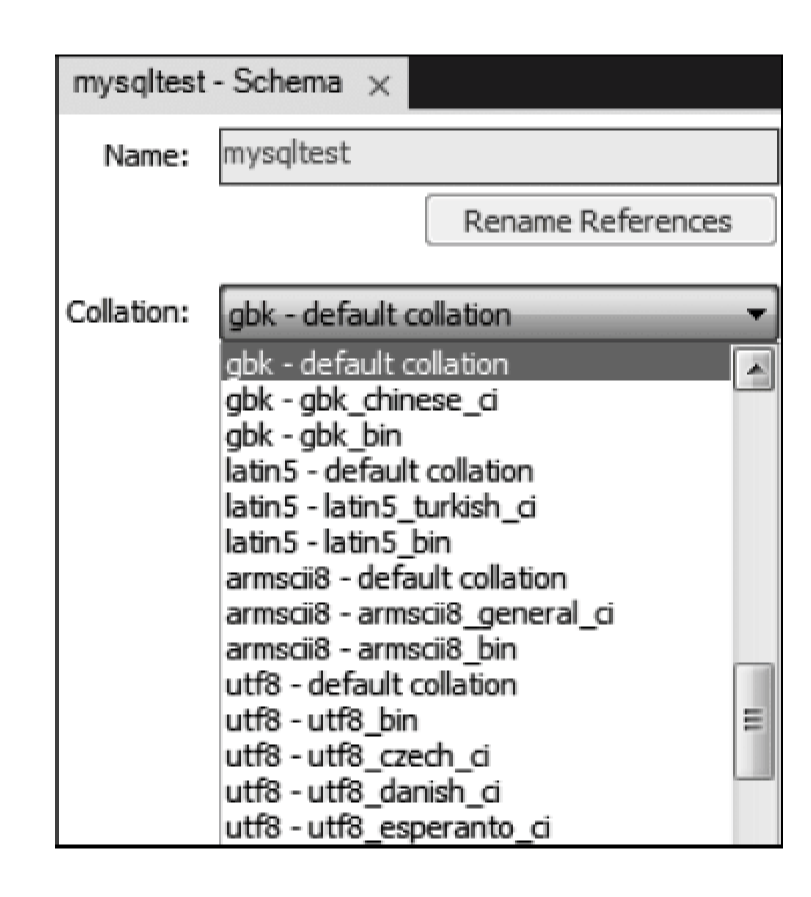


图 3-2　选择字符集

② 进入设置数据库参数界面。输入数据库名，选择字符集和排序规则，再单击 Apply 按钮。

③ 进入创建数据库脚本显示对话框，再单击 Apply 按钮，即可完成数据库的创建。

④ 在数据库操作界面（图 3-3）的 schemas 区域，就可以看到刚才创建的数据库 mysqltest，还有前面创建的数据库 teaching 以及 MySQL 中的系统数据库了。

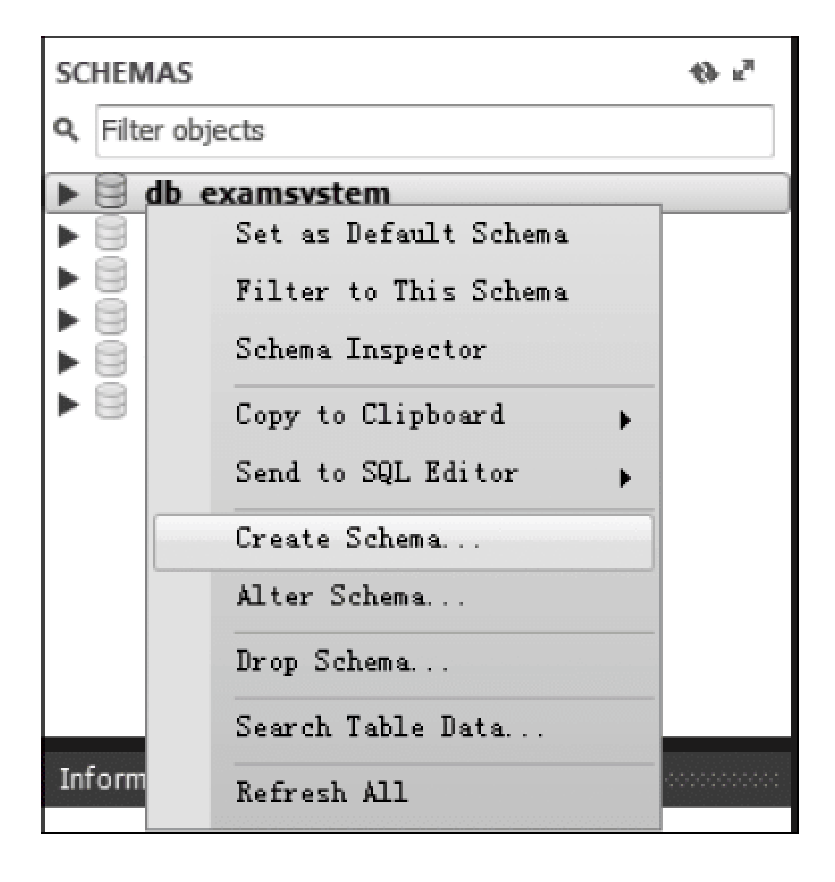


图 3-3　数据库操作主界面

（3）简述 MySQL 数据库的特点。

答：MySQL 与其他的大型数据库例如 Oracle、DB2、SQL Server 等相比功能稍弱一些，其主要有以下特点：

- 具有强大的数据保护功能，可以处理拥有万亿条记录的海量数据。
- 支持常见的 SQL 语句规范。
- 可移植性高，安装简单。
- 具有良好的运行效率，有丰富信息的网络支持。
- 相对其他大型数据库，调试、管理、优化简单。
- 可移植性、可扩展性和高灵活性。
- 提供用于管理、检查、优化数据库操作的管理工具。

（4）存储引擎 InnoDB、MyISAM 和 MEMORY 各有什么优缺点？

答：MySQL 数据库提供了多种存储引擎，用户可以根据不同的需求为数据表选择不同的存储引擎。MySQL 5.7 选择 InnoDB 作为默认存储引擎。InnoDB 是事务型数据库的首选引擎，是具有提交、回滚和崩溃恢复能力的事务安全存储引擎，支持行锁定和外键约束。

InnoDB 是 MySQL 上第一个提供外键约束的表引擎，而且 InnoDB 对事务处理的能力，也是 MySQL 其他存储引擎所无法与之比拟的。相对 MEMORY 来说，写处理能力差些，且会占用较多磁盘空间以保留数据和索引。InnoDB 存储引擎中支持自动增长列 auto_increment。

MEMORY 存储引擎是 MySQL 中的一类特殊的存储引擎。其使用存储在内存中的内容来创建表，而且所有数据也放在内存中。这些特性都与 InnoDB 存储引擎、MyISAM 存储引擎不同。

MyISAM 存储引擎曾是 MySQL 的默认存储引擎。MyISAM 存储引擎不支持事务、外键约束，但访问速度快，对事务完整性不要求，适合于以 select/insert 为主的表。

现在的 MyISAM 增加了很多有用的扩展。MyISAM 存储引擎的表存储成三个文件。

3. 上机练习题

题目要求：登录数据库系统以后，分别创建 student 和 teacher 数据库，数据库都创建成功后，删除 teacher 数据库。然后查看数据库系统中还存在哪些数据库。

主要实现过程如下所示：

（1）登录 MySQL 数据库客户端；

（2）查看数据库系统中已存在的数据库；

（3）查看该数据库系统支持的存储引擎的类型；

(4) 创建 student 数据库和 teacher 数据库；

(5) 再次查看数据库系统中已经存在的数据库，确保 student 和 teacher 数据库已经存在；

(6) 删除 teacher 数据库；

(7) 再次查看数据库系统中已经存在的数据库，确保 teacher 数据库已经删除。

答：命令和运行结果分别显示如下：

(1)
```
C:\Users\Administrator>mysql -u root -p
Enter password: ******
```

(2)
```
mysql> show databases;
        +------------------+
        | Database         |
        +------------------+
        | information_schema |
        | db_examsystem    |
        | mysql            |
        | mysqltest        |
        | performance_schema |
        | sakila           |
        | sys              |
        | teaching         |
        | world            |
        +------------------+
        9 rows in set (0.17 sec)
```

(3)
```
mysql> show engines; -- 结果参看教材第 3.5.1 小节
```

(4)
```
mysql> create database if not exists teacher;
        Query OK, 1 row affected (0.07 sec)
mysql> create database if not exists student;
        Query OK, 1 row affected (0.02 sec)
```

(5)
```
mysql> show databases;
        +------------------+
        | Database         |
        +------------------+
        | information_schema |
        | db_examsystem    |
        | mysql            |
        | mysqltest        |
        | performance_schema |
        | sakila           |
        | student          |
        | sys              |
        | teacher          |
        | teaching         |
        | world            |
        +------------------+
        11 rows in set (0.00 sec)
```

(6)
```
mysql> drop database teacher;
        Query OK, 0 rows affected (0.28 sec)
```

（7）mysql > show databases;

```
+------------------+
| Database         |
+------------------+
| information_schema |
| db_examsystem    |
| mysql            |
| mysqltest        |
| performance_schema |
| sakila           |
| student          |
| sys              |
| teaching         |
| world            |
+------------------+
10 rows in set (0.00 sec)
```

第 4 章　表及数据完整性

4.1　教学要求

4.1.1　基本要求

(1) 了解 InnoDB 存储引擎表空间的基本概念。

(2) 掌握创建数据库表的命令格式，以及表中各字段的数据类型和取值范围。

(3) 熟练掌握数据表的创建、查看、修改和删除等操作。

(4) 熟练掌握数据表记录的添加、修改和删除操作。

(5) 掌握利用 MySQL Workbench 管理表的基本过程。

(6) 了解表的数据完整性的用途。

(7) 掌握数据表的各种约束的创建和管理操作。了解表的数据完整性规则与数据的正确性、精确性和可靠性的关系。

(8) 了解临时表的用途和管理操作。

4.1.2　重点与难点

1. 本章重点

(1) 创建数据库表的命令格式。

(2) 数据表的创建、查看、修改和删除等操作。

(3) 表记录的添加、修改和删除操作。

(4) 掌握利用 MySQL Workbench 管理表的基本过程。

(5) 掌握数据表的各种约束的创建和管理操作。

2. 本章难点

(1) 创建数据库表的命令格式。

(2) 数据表的创建、查看、修改和删除等操作。

(3) 表的数据完整性的用途。

(4) 数据表的各种约束的创建和管理操作。

4.2　习题参考答案

1. 选择题

(1) 要快速完全清空一张表中的记录可使用如下语句________。

A. truncate table B. delete table

C. drop table D. clear table

(2) 使用 insert 命令插入记录时，使用________关键字会忽略导致重复关键字的错误记录。

A. no same B. ignore C. repeat D. unique

(3) SQL 语句中修改表结构的命令是________。

A. modify table B. modify structure

C. alter table D. alter structure

(4) 只修改列的数据类型的指令是________。

A. alter table … alter column B. alter table … modify column…

C. alter table … update … D. alter table … update column…

(5) 删除列的命令是________。

A. alter table … delete… B. alter table … delete column…

C. alter table … drop … D. alter table … drop column…

(6) 创建表时，不允许某列为空可以使用________命令。

A. not null B. no null C. not blank D. no blank

(7) 以下________指令无法增加记录。

A. insert into … values … B. insert into … select…

C. insert into … set … D. insert into … update…

(8) 关于 truncate table 描述不正确的是________。

A. truncate 将删除表中的所有数据

B. 表中包含 auto_increment 列，使用 truncate table 可以重置序列值为该列初始值

C. truncate 操作比 delete 操作占用资源多

D. truncate table 删除表，然后重新构建表

(9) 在创建表时，可以使用________关键字使当前建立的表为临时表。

A. ignore B. temporary C. temptable D. truncate

(10) 下列描述正确的是________。

A. 一个数据库只能包含一个数据表 B. 一个数据库可以包含多个数据表

C. 一个数据库只能包含两个数据表 D. 一个数据表可以包含多个数据库

本题答案：A B C B D A D C B B

2. 简答题

(1) 简述在创建表结构时，常用哪些数据类型？其主要功能是什么？

答：在创建 MySQL 表结构时，常用的 4 类标准数据类型的主要作用是：

① 字符串类型用于存储普通文本值或特殊类型 set 和 enum 类型。

② datetime 数据类型存储日期和时间，date 数据类型存储日期，time 数据类型存储时间。Timestamp 存储时间标签，year 用于存储年份。

③ 数字类型包括整数类型和浮点类型，整数类型是数据库中最基本的数据类型，用于存储各种数值范围的整数。浮点类型用于存储定点小数或近似数字数据类型。

④ 二进制类型是在数据库中存储二进制数据的数据类型。

(2) 简述创建表时各类约束对表中数据的作用。

答：在定义表结构时，还可以定义与该表相关的完整性约束条件，包括实体完整性、参照完整性和用户自定义完整性。这些约束条件主要包括 not null(非空约束)、primary key(主键约束)、unique(唯一性约束)、foreign key(外键参照完整性约束)以及 check(检查约束)。各种约束对表中数据的作用是：

① 检查约束限制一列的允许值。

② 非空约束限制在表的一个列中必须输入一个值。

③ 唯一约束确保一列或列的组合没有重复值。

④ 主键确保一个表中的每一行可以被指定列或指定列的组合唯一地标识。一个表只能有一个主键，但可以创建多个唯一约束。

⑤ 外键强制一个列只能取一个被引用的表中存在的值。

(3) MySQL 支持的数据完整性有哪几类？各有什么作用？

答：数据完整性可以根据其作用的数据库对象和范围不同分为 4 类。具体内容包括：

① 域完整性。要求列的数值具有正确的类型、格式和有效值范围，并确定是否允许有空值。通常使用有效性检查强制域完整性，也可以通过限定列中允许的数据类型、格式或有效值范围来强制数据完整性。

② 实体完整性。要求表中的每一行具有唯一的标识。在数据库中，如 student 表中的列 studentno 被设为主键，则会保证每个学生只有一个学号，而且是唯一的。

③ 引用完整性。指两个表的主键与外键之间定义的数据完整性，将确保主键和外键的关系。引用完整性可以保证两个引用表间的数据的一致性。

④ 用户定义完整性。用户可以根据其应用环境的不同，对数据库设置一些特殊的约束条件，反映某一具体应用所涉及的数据必须满足的语句要求。用户定义完整性的实现机制有规则(Rule)、触发器(Trigger)和存储过程(Stored Procedure)及创建数据表时的所有约束(Constraint)。

(4) 简述在 MySQL Workbench 中创建含有主键的表的步骤。

答：创建 MySQL 数据表，可以利用 MySQL Workbench 实现可视化方式。下面介绍利用可视化方式创建表的过程。

① 打开 MySQL Workbench 工具，在 Schemas 区域找到数据库列表。

② 选择一个数据库为当前数据库，右击该数据库，在弹出的菜单中执行 Set as Default Schema 命令，该数据库就变成当前数据库。

③ 选择 Tables 选项并右击，在弹出的菜单中执行 Create Table 命令。进入数据表创建初始界面。

④ 输入数据库表名，选择默认数据库引擎 InnoDB，参照设计好的表结构，分别输入列名，选择数据类型，输入不同类型的参数或默认值。

⑤ 单击 Apply 按钮，进入脚本审核对话框。单击 Show Logs 按钮，可以进一步查看脚本。单击 Finish 按钮，完成数据表的创建。

(5) 简述在 MySQL Workbench 中修改表数据的步骤。

答：数据表创建成功后，就可以实现对数据表结构进行修改、删除等操作，也可以编辑、查询和添加数据。

① 选择一个表，执行 Select Rows-Limit 1000 命令，就可以进入查询窗体。在该窗体中，不但可以进行数据添加，还可以执行查询、复制数据操作，也可以修改表结构和删除表格等操作。读者可以对相关工具栏按钮进行数据的添加删除操作。

② 单击 Apply 按钮后，再单击 Show Logs 按钮，可以进一步查看插入数据的脚本。

③ 单击 Finish 按钮，完成数据表数据的输入过程。

同样，应用可视化 MySQL Workbench 工具可以对数据表进行利用命令实现添加、删除、修改和查询等各种操作。

3. 上机练习题

(1) 创建 booksmgt 数据库，在 booksmgt 数据库中使用 MySQL 语句创建表 book 和表 author，结构如下：

```
book(bookid  char(6),bookname  varchar(30),  price float(5,2))
author(authorid char(6), authorname  varchar(10), bookid  char(6), phone varchar(15))
```

设置 book 中的 bookid 为主键，author 表中的 bookid 为外键。

答：① 创建表 book 的 MySQL 语句：

```
mysql> create database booksmgt;
       Query OK, 1 row affected (0.19 sec)
mysql> use  booksmgt;
       Database changed
mysql> create table book  (
    -> bookid char(6) not null,
    -> bookname varchar(30) null,
    -> price float(5, 2) null,
    -> constraint PKbook primary  key ( bookid asc  )
    -> );
       Query OK, 0 rows affected (0.78 sec)
```

② 创建表 author 的 MySQL 语句：

```
mysql> create table author(
    -> anthor_name nchar(4) not null,
    -> bookid nchar(6) not null,
    -> address nchar(30) not null
    -> ) ;
       Query OK, 0 rows affected (0.18 sec)
```

③ 设置 book 中的 bookid 为主键，author 表中的 bookid 为外键：

```
mysql> alter table author
    -> add constraint FK_book_author
    -> foreign key(bookid) references book(bookid);
       Query OK, 0 rows affected (0.42 sec)
       Records: 0  Duplicates: 0  Warnings: 0
```

(2) 在 booksmgt 数据库中利用 MySQL 语句创建一个图书销售表 booksales 结构如下：booksales(bookid nchar(6)，sellnum int，selldate datetime)。分别利用 insert、delete、update 语句添加、删除和更新数据。

答：① 利用 MySQL 语句创建表 booksales 的代码。

```
mysql > create table booksales(
    -> bookid nchar(6) not null,
    -> sellnum int not null,
    -> selldate datetime not null
    -> ) ;
       Query OK, 0 rows affected (0.14 sec)
```

② 利用 insert 语句为表 booksales 添加数据：

```
mysql > Insert into booksales values ('m00011',7, '2016 - 12 - 20');
       Query OK, 1 row affected (0.12 sec)
mysql > Insert into booksales(bookid,sellnum,selldate)
    -> values ('m00018',3, '2016 - 11 - 16');
       Query OK, 1 row affected (0.07 sec)
```

③ 利用 update 语句为表 booksales 更新数据：

```
mysql > update booksales  set sellnum = 11
    -> where bookid = 'm00011';
       Query OK, 1 row affected (0.09 sec)
```

④ 利用 delete 语句删除表 booksales 的数据：

```
mysql > delete from  booksales
    -> where  bookid = 'm00011';
       Query OK, 1 row affected (0.08 sec)
```

(3) 利用 MySQL 语句先删除表 booksales 中销售时间在 2016 年 12 月以前的记录，再删除全部记录，然后删除该表。

答：利用 MySQL 语句删除表 booksales 数据的代码如下：

```
mysql > Delete from  booksales
    -> where selldate <'2016 - 12 - 01';
       Query OK, 1 row affected (0.04 sec)
mysql > truncate  table booksales; -- 删除所有数据
       Query OK, 0 rows affected (0.16 sec)
```

附：本书中常用的数据库 teaching 中的表结构和表记录。

(1) 学生信息表 student

```
# student 表结构
create table if not exists student (
studentno  char(11) not null comment'学号',
sname char(8) not null comment'姓名',
sex enum('男', '女') default '男' comment'性别',
birthdate date not null comment'出生日期',
entrance int(3)  null comment'入学成绩',
```

```
phone varchar(12) not null comment'电话',
Email varchar(20) not null comment'电子信箱',
primary key (studentno)
);
```

#student 表常用数据

studentno	sname	sex	birthdate	entrance	phone	Email
18122210009	许东山	男	1999/11/5	789	13623456778	qwe@163.com
18122221324	何白露	女	2000/12/4	879	13178978999	heyy@sina.com
18125111109	敬横江	男	2000/3/1	789	15678945623	jing@sina.com
18125121107	梁一苇	女	1999/9/3	777	13145678921	bing@126.com
18135222201	凌浩风	女	2001/10/6	867	15978945645	tang@163.com
18137221508	赵临江	男	2000/2/13	789	12367823453	ping@163.com
19111133071	崔依歌	女	2001/6/6	787	15556845645	cui@126.com
19112100072	宿沧海	男	2002/2/4	658	12545678998	su12@163.com
19112111208	韩山川	男	2001/2/14	666	15878945612	han@163.com
19122203567	封月明	女	2002/9/9	898	13245674564	jiao@126.com
19123567897	赵既白	女	2002/8/4	999	13175689345	pingan@163.com
19126113307	梅惟江	女	2003/9/7	787	13245678543	zhu@163.com

(2) 课程信息表 course

```
#course 表结构
create table if not exists course (
courseno  char(6) not null,
cname  char(6) not null,
type char(8) not null,
period int(2) not null,
exp int(2) not null,
term int(2) not null,
primary key (courseno)
);
```

#course 表常用数据

Courseno	cname	type	period	exp	term
c05103	电子技术	必修	64	16	2
c05109	C 语言	必修	48	16	2
c05127	数据结构	必修	64	16	2
c05138	软件工程	选修	48	8	5
c06108	机械制图	必修	60	8	2
c06127	机械设计	必修	64	8	3
c06172	铸造工艺	选修	42	16	6
c08106	经济法	必修	48	0	7
c08123	金融学	必修	40	0	5
c08171	会计软件	选修	32	8	8

(3) 成绩信息表 score

```
#score 表结构
create table if not exists score(
studentno  char(11) not null,
courseno  char(6) not null,
daily float(3,1) default 0,
final float(3,1) default 0,
```

```
primary key (studentno , courseno)
);
#score 表常用数据
studentno      courseno  daily  final
18122210009    c05103    87     82
18122210009    c05109    77     91
18122221324    c05103    88     62
18122221324    c05109    91     77
18125111109    c08106    79     99
18125111109    c08123    85     92
18125111109    c08171    77     92
18125121107    c05103    74     91
18125121107    c05109    89     62
18135222201    c05109    99     92
18135222201    c08171    95     82
18137221508    c08106    78     95
18137221508    c08123    78     89
18137221508    c08171    88     98
19111133071    c05103    82     69
19111133071    c05109    77     82
19112100072    c05109    87     86
19112100072    c06108    97     97
19112111208    c05109    85     91
19112111208    c06108    89     95
19122111208    c06127    78     67
19122203567    c05103    65     98
19122203567    c05108    88     89
19122203567    c06127    79     88
19123567897    c05103    85     77
19123567897    c06127    99     99
19126113307    c06108    66     82
19126113307    c08171    88     79
```

(4) 教师信息表 teacher

```
#teacher 表结构
create table if not exists teacher (
teacherno  char(6) not null comment '教师编号',
tname  char(8) not null comment'教师姓名',
major  char(10) not null comment '专业',
prof char(10) not null comment '职称',
department char(16) not null comment '部门',
primary key (teacherno)
);
#teacher 表常用数据
teacherno  tname   major     prof    department
t05001     苏超然  软件工程  教授    计算机学院
t05002     常杉    会计学    助教    管理学院
t05003     孙释安  网络安全  教授    计算机学院
t05011     卢敖治  软件工程  副教授  计算机学院
t05017     茅佳峰  软件测试  讲师    计算机学院
```

```
t06011        夏南望    机械制造    教授      机械学院
t06023        葛庭宇    铸造工艺    副教授    材料学院
t07019        韩既乐    经济管理    讲师      管理学院
t08017        时观      金融管理    副教授    管理学院
```

(5) 纽带表 teach_course

```
#teach_course 表结构
create table if not exists teach_course (
teacherno char(6) not null,
courseno  char(6) not null,
primary key (teacherno,courseno)
);
#teach_course 表常用数据
teacherno        courseno
t05001           c05109
t05002           c05127
t05003           c05127
t05011           c05138
t05017           c05127
t06011           c06127
t06023           c06172
t07019           c08123
t08017           c08106
```

(6) 选课信息表 se_course

```
#se_course 表结构
create table se_course
(sc_no int(6) not null auto_increment,
studentno  char(11) not null,
courseno  char(6) not null,
teacherno char(6) not null,
sc_time timestamp not null default now(),
primary key (sc_no)
);
#se_course 表常用数据
sc_no    studentno    courseno teacherno   score   sc_time
   1    19120000111    co1236    t01237      NULL    2017-09-01 18:40:23
   2    19120000222    co1237    t01239      NULL    2017-09-05 18:24:22
```

第5章 数据检索

5.1 教学要求

5.1.1 基本要求

(1) 了解选择、投影和连接的基本概念。

(2) 掌握 select 语句的基本格式和各个子句的格式。

(3) 掌握单表查询中 where、group by、order by、having 和 limit 的功能。

(4) 掌握利用 group by 子句与 with rollup 选项进行统计操作时，各个聚合函数的格式、功能和基本应用。

(5) 了解多表连接的工作过程，掌握 4 种多表连接的格式和应用。

(6) 了解子查询的工作过程和应用场合。

(7) 掌握子查询在 select、from 和 where 子句中实现的操作。

(8) 掌握利用子查询成批添加、删除和修改表数据的操作方法。

(9) 掌握使用正则表达式进行模糊查询的格式和方法。

5.1.2 重点与难点

1. 本章重点

(1) select 语句的基本格式和各个子句的格式。

(2) 单表查询中 where、group by、order by、having 和 limit 各子句的功能。

(3) 聚合函数的格式、功能和基本应用。

(4) 内连接的格式和基本操作。

(5) 子查询在 select、from 和 where 子句中实现的操作。

(6) 利用子查询成批添加、删除和修改表数据的操作方法。

(7) 使用正则表达式进行模糊查询的格式和方法。

2. 本章难点

(1) 在 where 子句中使用 like、in、between 关键字时的各种操作。

(2) 利用 group by 子句与 with rollup 选项进行统计操作。

(3) 外连接的格式和应用场合。

(4) select 语句中使用子查询时的技巧。

(5) 使用正则表达式进行模糊查询的格式和方法。

5.2　习题参考答案

1. 选择题

(1) select 语句中使用________关键字可以将重复行屏蔽。

A. order　by　　B. having　　C. top　　D. distinct

(2) 在 select 语句中，可以使用________子句，将结果集中的数据行根据选择列的值进行逻辑分组，以便能汇总表内容的子集，即实现对每个组的聚集计算。

A. limit　　B. group by　　C. where　　D. order by

(3) 使用空值查询时，表示一个列 RR 不是空值的表达式是________。

A. RR is null　　B. RR = null　　C. RR <> null　　D. RR is not null

(4) select * from city limit 5,10 描述正确的是________。

A. 获取第 6 条到第 10 条记录　　B. 获取第 5 条到第 10 条记录

C. 获取第 6 条到第 15 条记录　　D. 获取第 5 条到第 15 条记录

(5) select 语句中用于实现关系的选择运算的短语是________。

A. for　　B. while　　C. where　　D. condition

(6) 关于 select 语句以下哪一个描述是错误的？________

A. select 语句用于查询一个表或多个表的数据

B. select 语句属于数据操作语言(DML)

C. select 语句的输出列必须是基于表的列

D. select 语句表示数据库中一组特定的数据记录

(7) select 语句的执行过程是从数据库中选取匹配的特定记录和字段，并将这些数据组织成一个结果集，然后以________的形式返回。

A. 结构体数组　　B. 系统表　　C. 永久表　　D. 临时表

(8) 现有订单表 orders，包含用户信息 userid，产品信息 productid，以下________语句能够返回至少被订购过两回的 productid。

A. select productid from orders where count(productid)>1

B. select productid from orders where max(productid)>1

C. select productid from orders where having count(productid)>1 group by productid

D. select productid from orders group by productid having count(productid)>1

本题答案：D B D C C C D D

2. 思考题

(1) 简述 select 语句的各个子句的作用。

答：select 语句是 SQL 语言从数据库中获取信息的一个基本语句，其各子句的作用简述如下：

- select 用于描述结果集的列。

- from 用于指定所要查询数据源，如表、视图、表达式等。
- where 用于定义源表中的行要满足 select 语句的要求所必须达到的条件。
- group by 用于利用对查询结构的指定值进行分组统计。
- having 子句是应用于分组结果集的附加条件。
- order by 用于对查询结果进行排序。
- limit 用于限制查询的输出结果行。使用 limit 关键字来限制查询结果的行数是 MySQL 数据库的特色。

(2) MySQL 中通配符与正则表达式的区别是什么?

答：使用通配符和正则表达式都可以进行模糊查询，MySQL 中有两种通配符："%"和下画线"_"。"%"可以匹配一个或多个字符，可以代表任意长度的字符串，长度可以为 0；"_"只匹配一个字符。一般在查询中仅仅用于简单的字符串匹配。

正则表达式通常用来检索或替换符合某个模式的文本内容，根据指定的匹配模式匹配文本中符合要求的特殊字符串。例如从一个文本文件中提取电话号码，查找一篇文章中重复的单词或者替换用户输入的某些词语等。正则表达式强大而且灵活，可以应用于非常复杂的查询。

(3) 说明在 select 语句中使用聚合函数应该注意的问题。

答：聚合函数用于对一组值进行计算并返回一个单一的值。除 count 函数之外，其他聚合函数忽略空值。聚合函数经常与 select 语句的 group by 子句一同使用，select 语句的输出项中只能包含 group by 子句分组列和聚合函数。聚合函数的作用是在结果集中通过对被选列值的收集处理，返回一个数值型的计算结果。

(4) 将 null 与其他值比较会产生什么结果? 数值列中存在 null 会产生什么结果?

答：MySQL 中涉及空值的查询用 null 来表示。null 不同于数据库中的其他任何值。where 子句通常会返回比较的计算结果为真的行。为了取得列中含有 null 的行，MySQL 语句包含了操作符功能 is [not] null。

一个字段值是空值或者不是空值，要表示为"is null"或"is not null"。不能表示为"=null"或"<> null"。如果写成"字段=null"或"字段<> null"，系统的运行结果都直接处理为 null 值，按照 false 处理而不报错。

(5) 简述连接查询和利用 union 语句合并结果集的应用区别。

答：连接查询是通过 join 操作将多个表按照指定方式连接成一个新的中间表，然后再进行查询操作。union 操作是将多个 select 语句的返回结果组合到一个结果集中。当要检索的数据在不同的结果集中，并且不能够利用一个单独的查询语句得到时，可以使用 union 合并多个结果集。将两个或更多查询的结果合并为单个结果集，该结果集包含联合查询中的所有查询的全部行。union 运算不同于使用连接合并两个表中的列的运算。

(6) 什么情况下使用 limit 来限制查询结果的数量?

答：使用 limit 关键字来限制查询结果的条数是 MySQL 数据库的特色。当输出结果中的记录行较多时，可以使用 limit 子句将符合指定范围的记录行过滤出来。

3. 上机练习题(本题利用 teaching 数据库进行操作)

(1) 查询 course 表中的所有记录。

答：程序代码如下：

```
select * from course;
```

(2) 查询 student 表中女生的人数。

答：程序代码如下：

```
select count( * ) from student
where  sex = '女';
```

(3) 查询 teacher 表中每一位教授的教师号、姓名和专业名称。

答：代码和运行结果如下：

```
mysql > select teacherno,tname, major from teacher
    -> where  prof = '教授';
     +----------+-------+-------+
     | teacherno |  tname  | major   |
     +----------+-------+-------+
     | t05001    | 苏超然 | 软件工程 |
     | t05003    | 孙释安 | 网络安全 |
     | t06011    | 夏南望 | 机械制造 |
     | t06027    | 陶期年 | 纳米技术 |
     +----------+-------+-------+
     4 rows in set (0.15 sec)
```

(4) 按性别分组，求出 student 表中每组学生的平均年龄。

答：代码和运行结果如下：

```
mysql > select sex,avg(year(now()) - year(birthdate))
    -> from student
    -> group by  sex;
     +----+---------------------------+
     | sex  | avg(year(now()) - year(birthdate))  |
     +----+---------------------------+
     | 男   |                    17.0000        |
     | 女   |                    15.8571        |
     +----+---------------------------+
     2 rows in set (0.00 sec)
```

(5) 创建新表，新表中包括学号、学生姓名、课程号和总评成绩。其中，总评成绩＝final＊0.8＋daily＊0.2。

答：代码和运行结果如下：

```
mysql > create table stu_totall
    -> select student.studentno,student.sname,score.courseno,
    ->      final * 0.8 + daily * 0.2 as 'totall'
    -> from student,score
    -> where student.studentno = score.studentno;
     Query OK, 26 rows affected (0.54 sec)
     Records: 26  Duplicates: 0  Warnings: 0
mysql > select * from stu_totall;
     +----------+-------+-------+------+
     | studentno    | sname   | courseno | totall |
```

```
+-----------+--------+--------+-------+
| 19111133071 | 崔依歌   | c05103    |    70.0 |
| 19111133071 | 崔依歌   | c05109    |    79.5 |
| 19122203567 | 封月明   | c05103    |    90.1 |
| 19122203567 | 封月明   | c05108    |    87.0 |
| 19122203567 | 封月明   | c06127    |    86.2 |
| 19112111208 | 韩山川   | c05109    |    88.1 |
…
| 19123567897 | 赵既白   | c06127    |    99.0 |
| 18137221508 | 赵临江   | c08106    |    91.6 |
| 18137221508 | 赵临江   | c08123    |    86.8 |
| 18137221508 | 赵临江   | c08171    |    96.0 |
+-----------+--------+--------+-------+
26 rows in set (0.00 sec)
```

(6) 统计每个学生的期末成绩平均分。

答：代码和运行结果如下：

```
mysql> select student.studentno,student.sname,avg(score.final)
    -> from student join score
    -> on student.studentno = score.studentno
    -> group by score.studentno ;
+----------+------+-----------------+
| studentno   | sname  | avg(score.final)   |
+----------+------+-----------------+
| 18122210009 | 许东山 |        86.50000 |
| 18122221324 | 何白露 |        69.50000 |
| 18125111109 | 敬横江 |        94.33333 |
| 18125121107 | 梁一苇 |        76.50000 |
| 18135222201 | 凌浩风 |        87.00000 |
| 18137221508 | 赵临江 |        94.00000 |
| 19111133071 | 崔依歌 |        75.50000 |
| 19112111208 | 韩山川 |        93.00000 |
| 19122203567 | 封月明 |        91.66667 |
| 19123567897 | 赵既白 |        99.00000 |
| 19126113307 | 梅惟江 |        93.50000 |
+----------+------+-----------------+
11 rows in set (0.03 sec)
```

(7) 输出 student 表中年龄最大的男生的所有信息。

答：代码和运行结果如下：

```
mysql> select * from student st
    -> where st.birthdate =
    ->       ( select min(stu.birthdate) from student stu
    ->         where stu.sex = '男');
+----------+----+----+-----+-------+-------+-------+
|  studentno  |sname |sex |birthdate | entrance|phone      |Email    |
+----------+----+----+-----+-------+-------+-------+
|18122210009|许东山| 男|1999-11-05|789|13623456778 |qwe@163.com|
+----------+----+----+-----+-------+-------+-------+
1 row in set (0.06 sec)
```

(8) 查询 teacher 表中没有职称的职工的教师号、姓名、专业和部门。

答：代码和运行结果如下：

```
mysql> select teacherno, tname, major, department
    -> from teacher
    -> where prof is null;
      Empty set (0.03 sec)
```

第6章 索引和视图

6.1 教学要求

6.1.1 基本要求

(1) 了解索引的分类和用途。主要了解索引和表的关系。
(2) 掌握索引的创建、修改、删除等基本操作。
(3) 学会利用 MySQL Workbench 工具创建和管理索引。
(4) 掌握视图的概念、功能和用途,了解索引和视图的关系。
(5) 掌握视图的创建、修改、删除等基本操作。
(6) 掌握利用视图添加、删除、修改和浏览表数据的基本原则和基本操作。
(7) 学会利用 MySQL Workbench 工具创建和管理视图。
(8) 掌握检查视图的应用。

6.1.2 重点与难点

1. 本章重点

(1) 索引的创建、修改、删除等基本操作。
(2) 视图的概念、功能和用途,了解索引和视图的关系。
(3) 掌握视图的创建、修改、删除等基本操作。
(4) 掌握利用视图添加、删除、修改和浏览表数据的基本原则和基本操作。
(5) 学会利用 MySQL Workbench 工具创建和管理视图。

2. 本章难点

(1) 索引和表的关系的理解。
(2) 利用视图添加、删除和修改表数据的条件。
(3) 利用视图添加、删除和修改表数据的基本操作。
(4) 掌握检查视图的应用。

6.2 习题参考答案

1. 选择题

(1) 下列________语句不能用于创建索引。

A. create index　　　　B. create table

C. alter table D. create database

(2) 下面对索引的相关描述正确的是________。

A. 经常被查询的列不适合建索引 B. 小型表适合建索引

C. 有很多重复值的列不适合建索引 D. 是外键或主键的列不适合建索引

(3) MySQL中不可对视图执行的操作有________。

A. select B. insert C. delete D. create index

(4) 对视图的描述错误的是________。

A. 视图是一张虚拟表

B. 视图定义包含limit子句时才能设置排序规则

C. 可以像查询表一样来查询视图

D. 被修改数据的视图只能是一个基表的列

(5) with check option属性对视图有________的用途。

A. 进行权限检查 B. 进行删除监测 C. 进行更新监测 D. 进行插入监测

(6) 索引可以提高________操作的效率。

A. insert B. update C. delete D. select

(7) 在MySQL中唯一索引的关键字是________。

A. fulltext B. only C. unique D. index

本题答案:D C D D A D C

2. 思考题

(1) 简述创建索引的必要性。

答:创建索引可以大大提高系统的查询性能,具体表现如下:

① 加快数据的检索速度。

② 通过创建唯一性索引,可以保证每一行数据的唯一性。

③ 加速表与表之间的连接,特别是在实现数据的参照完整性方面具有特别的意义。

④ 在使用order by和group by子句进行数据检索时,可以显著减少查询中分组和排序的时间。

⑤ 通过使用索引,可以在查询过程中使用查询优化器,提高系统性能。

(2) MySQL中普通索引、主键和唯一索引的区别是什么?

答:MySQL中最常用的索引有普通索引、唯一性索引和主键索引等类型。普通索引是MySQL中的基本索引类型,允许在定义索引的列中插入重复值和空值。主键索引是一种特殊的唯一索引,不允许有空值。一般是在建表的时候同时创建主键索引。一个表只能有一个主键索引。唯一索引中索引列的值必须唯一,允许有空值。如果是组合索引,则列值的组合必须唯一。在一个表上可以创建多个unique索引。

(3) 简述表和视图之间的关系。

答:视图是从一个或多个基本表导出的虚表,它的数据并不存储在数据库中,视图中的数据来自定义视图的查询所引用的表。视图的查询可像基本表一样参与数据库操作,但视图的更新最终落实到有关基本表的更新上。

(4) 简述创建视图的必要性。

答：创建视图的必要性分析如下：

① 将数据集中显示，简化用户的数据查询和处理。

② 简化数据操作，将数据库的设计和用户的使用分离开来，屏蔽数据库的复杂性。

③ 重新定制数据，使得数据便于共享。组合分区数据，可将来自不同表的多个查询结果组合成单一的结果集，利于用户查询和使用。

④ 简化用户权限的管理，增强了安全性。

3. 上机练习题(本题利用 teaching 数据库中的表进行操作)

(1) 在 course 表的 cname 列上创建索引 IDX_cname。

答：代码和运行结果如下：

```
mysql> use teaching;
        Database changed
mysql> create index IDX_cname on course(cname);
        Query OK, 0 rows affected (0.47 sec)
        Records: 0  Duplicates: 0  Warnings: 0
```

(2) 在 student 表的 studentno 和 phone 列上创建唯一索引 uq_stu，并输出 student 表中的记录，查看输出结果的顺序。

答：代码和运行结果如下：

```
mysql> create index unique uq_stu on student(studentno,phone);
        Query OK, 0 rows affected (0.16 sec)
        Records: 0  Duplicates: 0  Warnings: 0
mysql>  select studentno,phone,sname from  student;
         +----------+----------+------+
         | studentno  | phone           | sname  |
         +----------+----------+------+
         | 18122210009 | 13623456778  | 许东山 |
         | 18122221324 | 13178978999  | 何白露 |
         | 18125111109 | 15678945623  | 敬横江 |
         | 18125121107 | 132123456777 | 梁一苇 |
         | 18135222201 | 15978945645  | 凌浩风 |
         | 18137221508 | 12367823453  | 赵临江 |
         | 19111133071 | 15556845645  | 崔依歌 |
         | 19112111208 | 15878945612  | 韩山川 |
         | 19122203567 | 13245674564  | 封月明 |
         | 19123567897 | 13175689345  | 赵既白 |
         | 19126113307 | 13245678543  | 梅惟江 |
         +----------+----------+-----+
11 rows in set (0.00 sec)
```

(3) 创建一个视图 v_teacher，查询所有“计算机学院”教师的信息。

答：代码和运行结果如下：

```
mysql> create view  v_teacher
    -> as
    -> select *
```

```
    -> from teacher
    -> where department = '计算机学院';
       Query OK, 0 rows affected (0.07 sec)
mysql> select * from v_teacher;
       +---------+-------+--------+------+----------+
       | teacherno | tname | major | prof | department |
       +---------+-------+--------+------+----------+
       | t05001 | 苏超然 | 软件工程 | 教授 | 计算机学院 |
       | t05003 | 孙释安 | 网络安全 | 教授 | 计算机学院 |
       | t05011 | 卢敖治 | 软件工程 | 副教授 | 计算机学院 |
       | t05017 | 茅佳峰 | 软件测试 | 讲师 | 计算机学院 |
       +---------+-----+------+-----+---------+
4 rows in set (0.03 sec)
```

(4) 创建一个视图 v_avgstu,查询每个学生的学号、姓名及平均分,并且按照平均分降序排序。

答:代码和运行结果如下:

```
mysql> create view v_avgstu
    -> as
    -> select student.studentno, student.sname,
    ->       avg(score.final) as'average'
    -> from student, score
    -> where student.studentno = score.studentno
    ->       and score.final is not null
    -> group by student.studentno, student.sname
    -> order by avg(score.final) desc;
       Query OK, 0 rows affected (0.11 sec)
mysql> select * from v_avgstu;
       +---------+-------+--------+
       | studentno | sname | average |
       +---------+-------+--------+
       | 19123567897 | 赵既白 | 99.00000 |
       | 18125111109 | 敬横江 | 94.33333 |
       | 18137221508 | 赵临江 | 94.00000 |
       | 19126113307 | 梅惟江 | 93.50000 |
       | 19112111208 | 韩山川 | 93.00000 |
       | 19122203567 | 封月明 | 91.66667 |
       | 18135222201 | 凌浩风 | 87.00000 |
       | 18122210009 | 许东山 | 86.50000 |
       | 18125121107 | 梁一苇 | 76.50000 |
       | 19111133071 | 崔依歌 | 75.50000 |
       | 18122221324 | 何白露 | 69.50000 |
       +---------+-------+--------+
       11 rows in set (0.02 sec)
```

(5) 修改 v_teacher 的视图定义,添加 with check option 选项。

答:代码和运行结果如下:

```
mysql> alter view v_teacher
    -> as
```

```
    -> select * from teacher
    -> where department = '计算机学院'
    -> with check option ;
     Query OK, 0 rows affected (0.06 sec)
```

(6) 通过视图 v_teacher 向基表 teacher 中分别插入下列数据,并查看插入数据的情况。

('t05039','张馨月','计算机应用', '讲师', '计算机学院)
('t06018', '李书诚', '机械制造','副教授','机械学院')

答:代码和运行结果如下:

```
mysql> insert into v_teacher
    -> values('t05039','张馨月','计算机应用','讲师','计算机学院');
     Query OK, 1 row affected (0.06 sec)
mysql> insert into v_teacher
    ->   values('t06018','李书诚','机械制造','副教授','机械学院');
     ERROR 1369 (HY000): CHECK OPTION failed 'teaching.v_teacher'
mysql> select * from v_teacher;
     +---------+-------+------------+---------+------------+
     | teacherno | tname | major      | prof    |department|
     +---------+-------+----------+---------+----------+
     | t05001    | 苏超然 | 软件工程   | 教授     | 计算机学院 |
     | t05003    | 孙释安 | 网络安全   | 教授     | 计算机学院 |
     | t05011    | 卢敖治 | 软件工程   | 副教授   | 计算机学院 |
     | t05017    | 茅佳峰 | 软件测试   | 讲师     | 计算机学院 |
     | t05039    | 张馨月 | 计算机应用 | 讲师     | 计算机学院 |
     +---------+-------+----------+---------+----------+
     5 rows in set (0.00 sec)
mysql> select * from teacher;
     +----------+----+---------+------+---------+
     | teacherno | tname | major    | prof  |department|
     +----------+----+---------+------+---------+
     | t05001    | 苏超然 | 软件工程 | 教授  | 计算机学院 |
     | t05002    | 常杉   | 会计学   | 助教  | 管理学院   |
     | t05003    | 孙释安 | 网络安全 | 教授  | 计算机学院 |
     | t05011    | 卢敖治 | 软件工程 | 副教授 | 计算机学院 |
     | t05017    | 茅佳峰 | 软件测试 | 讲师  | 计算机学院 |
     | t05039    | 张馨月 | 计算机应用 | 讲师 | 计算机学院 |
     | t06011    | 夏南望 | 机械制造 | 教授  | 机械学院   |
     | t06023    | 葛庭宇 | 铸造工艺 | 副教授 | 材料学院   |
     | t06027    | 陶期年 | 纳米技术 | 教授  | 材料学院   |
     | t07019    | 韩既乐 | 经济管理 | 副教授 | 管理学院   |
     +----------+----+---------+------+---------+
     10 rows in set (0.00 sec)
```

(7) 通过视图 v_teacher 将基表 teacher 中教师编号为 t05039 的教师职称修改为“副教授”。

答:代码和运行结果如下:

```
mysql> update v_teacher
    -> set prof = '副教授'
```

```
    -> where teacherno = 't05039';
    Query OK, 1 row affected (0.07 sec)
    Rows matched: 1  Changed: 1  Warnings: 0
mysql>  select * from teacher
    -> where teacherno = 't05039';
    +---------+-------+----------+--------+----------+
    | teacherno | tname  | major      | prof    | department |
    +---------+-------+----------+--------+----------+
    | t05039    | 张馨月 | 计算机应用 | 副教授  | 计算机学院  |
    +---------+-------+----------+--------+----------+
    1 row in set (0.00 sec)
```

第7章 MySQL 编程基础

7.1 教学要求

7.1.1 基本要求

(1) 掌握系统变量的用途和使用方法。

(2) 掌握会话变量的定义方法和使用方法,了解 MySQL 中使用会话变量的机制,以及会话变量与局部变量的区别。

(3) 掌握定界符 delimiter、begin...end 语句块和预处理 SQL 语句的用途。

(4) 掌握创建、修改、删除和调用自定义函数的格式和方法。

(5) 了解 MySQL 中利用控制流语句编程的过程。

(6) 掌握 if 和 case 等条件控制语句的格式、功能和执行过程。

(7) 掌握 while、loop、repeat 等循环语句的格式、功能和执行过程。

(8) 学会编写较为复杂的自定义函数。

7.1.2 重点与难点

1. 本章重点

(1) 会话变量的定义方法和使用方法,以及会话变量与局部变量的区别。

(2) 定界符 delimiter、begin...end 语句块和预处理 SQL 语句的用途。

(3) 创建、修改、删除和调用自定义函数的格式和方法。

(4) if 和 case 等条件控制语句的格式、功能和执行过程。

(5) while、loop、repeat 等循环语句的格式、功能和执行过程。

(6) 编写较为复杂的自定义函数。

2. 本章难点

(1) MySQL 中使用会话变量的机制,以及会话变量与局部变量的区别。

(2) 预处理 SQL 语句的定义和使用。

(3) if 和 case 等条件控制语句的执行过程。

(4) while、loop、repeat 等循环语句的格式、功能和执行过程。

7.2 习题参考答案

1. 选择题

(1) 在 MySQL 语句中,可以匹配 0 个到多个字符的通配符是________。

A. * B. % C. ? D. -

(2) MySQL 提供的单行注释语句可以是使用________开始的一行内容。

A. /* B. # C. { D. /

(3) 在 MySQL 中会话变量前面的字符为________。

A. * B. # C. @@ D. @

(4) 若要计算表中数据的平均值,可以使用的函数是________。

A. sqrt B. avg C. square D. count

(5) ________语句用于执行预处理语句。

A. prepare B. deallocate C. execute D. using

本题答案:B B D B C

2. 思考题

(1) MySQL 的语言要素有哪些? 主要作用是什么?

答:MySQL 语言的要素及主要用途如下:

① 标识符,标识符是用来命名一些如数据库、表、视图、存储过程等数据库对象名称的字符序列,该序列由以字母或下画线开头的字母、数字或下画线(_)序列组成。

② 常量,表示一个特定数据值的符号,其格式取决于它所表示的值的数据类型。

③ 变量,一种是用户自己定义的局部变量和会话变量,另外一种是系统提供的全局变量。用于存储某过程中可以变化的值。

④ 注释,程序代码中不可执行的文本字符串,增加程序的可读性。

⑤ 运算符,用来执行算术运算、字符串连接、赋值以及在字段、常量和变量之间进行比较的操作符。

⑥ 函数,能够返回一个值的程序段。

⑦ 表达式,指符号和运算符的组合,其计算结果为单个数据值。

⑧ MySQL 语句,主要功能是创建和管理数据库对象、插入、检索、修改和删除数据,并实现服务器和客户端的操作和管理。

(2) 如何定义变量,如何给变量赋值?

答:定义局部变量需要使用 declare 命令,会话变量可以直接定义和赋值等,要在其名称前加上标志“@”,使用全局变量时,必须以标记符“@@”开头。

为变量赋值需要使用 set 和 select 命令。set 命令一次只能为一个变量赋值,select 命令可以同时为多个变量赋值。select 命令的赋值号为“ :=”。

(3) 流程控制语句包括哪些类型? 各自的作用是什么?

答:控制流语句主要包括 if...else 选择语句和 case 多分支语句,语句复合体 begin...

end 语句块,while、loop 和 repeat 等循环语句。MySQL 提供的控制流语句主要用于控制程序的顺序、选择和循环流程。

(4) MySQL 语句共分几类,各自的主要功能是什么?

答:MySQL 语句共分为 4 类,其主要作用如下:

① 数据定义语言,数据库对象的创建、修改和删除等都可以通过数据定义语言中的 create、alter、drop 等语句来完成。

② 数据操纵语言。用于检索和操作数据的 SQL 语句的子集。数据操纵语言是指用来查询、添加、修改和删除数据库中数据的语句,这些语句包括 select、insert、update、delete 等命令。

③ 数据控制语言,用来设置或更改数据库用户或角色权限的语句,包括 grant、revoke 等命令。

④ 控制流语句,用于控制 SQL 语句、语句块或者存储过程的执行流程。流程控制语句包括 begin…end、if…else、while、loop、return 等语句。

3. 上机练习题

(1) 利用预处理 SQL 语句输出数据库 teaching 中 teacher 表的前 5 行记录的部分数据。

答:代码和运行结果如下:

```
mysql> set @a=5;
       Query OK, 0 rows affected (0.00 sec)
mysql> prepare  pteac
    -> from "select * from  teacher  limit ?";
       Query OK, 0 rows affected (0.00 sec)
       Statement prepared
mysql> execute pteac using @a;
       +---------+-------+---------+--------+---------+
       | teacherno | tname  | major    | prof   |department |
       +---------+-------+---------+--------+---------+
       | t05001  | 苏超然 | 软件工程 | 教授   | 计算机学院 |
       | t05002  | 常杉   | 会计学   | 助教   | 管理学院   |
       | t05003  | 孙释安 | 网络安全 | 教授   | 计算机学院 |
       | t05011  | 卢敖治 | 软件工程 | 副教授 | 计算机学院 |
       | t05017  | 茅佳峰 | 软件测试 | 讲师   | 计算机学院 |
       +---------+-------+---------+--------+---------+
       5 rows in set (0.06 sec)
```

(2) 创建函数 casetwo,通过 case 语句首先判断传入参数的值是否为 7,如果条件成立则输出 1,如果条件不成立则再判断该传入参数的值是否为 14,如果成立则输出 2,当以上条件都不满足时输出-1。

答:代码和运行结果如下:

```
mysql> delimiter //
mysql> create function casetwo(x int)
    -> returns  int
    -> DETERMINISTIC
```

```
    -> begin
    -> case  x
    -> when 7  then set  x = 1;
    -> when 14  then set  x = 2;
    -> else  set  x = -1;
    -> end case;
    -> return  x;
    -> end //
       Query OK, 0 rows affected (0.03 sec)
mysql> delimiter ;
mysql> select casetwo (14);
       +---------+
       | casetwo(14) |
       +---------+
       |      2     |
       +---------+
       1 row in set (0.02 sec)
```

(3) 定义函数 exthree,应用 while 语句编程求 50 到指定整数的所有奇数之和。

答:代码和运行结果如下:

```
mysql> delimiter //
mysql> create  function  exthree(sn int) returns int
    -> DETERMINISTIC
    -> begin
    -> declare sum int default 0;
    -> declare n int default 0;
    -> set n = 51;
    ->   while  n <= sn  do
    -> set  sum = sum + n;
    -> set  n = n + 2;
    -> end while;
    -> return  sum;
    -> end //
  Query OK, 0 rows affected (0.03 sec)
mysql> delimiter ;
mysql> select exthree(150);
       +----------+
       | exthree(150) |
       +----------+
       |      5000   |
       +----------+
       1 row in set (0.00 sec)
```

(4) 定义函数 exfour,应用 while 语句和 iterate 语句求 100～150 的偶数之和。通过 leave 语句退出循环并输出结果。

答:代码和运行结果如下:

```
mysql> delimiter //
mysql> create  function exfour (n int)
     -> returns int
```

```
    -> DETERMINISTIC
    -> begin
    -> declare sum char(20) default 0;
    -> declare s int default 100;
    -> add_num: while  true  do
    -> set s = s + 2;
    -> if (s % 2 = 0) then
    -> set sum = sum + s;
    -> else
    -> iterate  add_num;
    -> end if;
    -> if(s = n) then
    -> leave  add_num;
    ->   end if;
    ->   end while add_num;
    ->   return  sum;
    ->   end //
   Query OK, 0 rows affected (0.03 sec)
mysql> delimiter ;
mysql> select exfour (150);
   +-----------+
   | exfour (150) |
   +-----------+
   |       3150  |
   +-----------+
   1 row in set (0.01 sec)
```

第 8 章 存储过程、游标和触发器

8.1 教学要求

8.1.1 基本要求

(1) 了解存储过程的功能,以及存储过程与游标、触发器和事件的关系。

(2) 掌握存储过程的创建、修改、删除和调用方法。

(3) 学会利用 MySQL Workbench 工具管理存储过程。

(4) 了解游标的作用和定义、使用方法。

(5) 了解触发器的功能,了解触发器和数据表的关系。

(6) 掌握触发器的创建、修改、删除和调用方法。

(7) 了解事件的功能,以及事件与触发器的区别。

(8) 掌握事件的创建和使用方法。

8.1.2 重点与难点

1. 本章重点

(1) 存储过程的功能,以及存储过程与游标、触发器和事件的关系。

(2) 存储过程的创建、修改、删除和调用方法。

(3) 游标的作用和定义、使用方法。

(4) 触发器的功能,了解触发器和数据表的关系。

(5) 触发器的创建、修改、删除和调用方法。

(6) 事件的创建和使用方法。

2. 本章难点

(1) 存储过程的创建、修改和调用方法。

(2) 利用游标访问结果集的方法。

(3) 触发器的创建、修改和调用方法。

(4) 事件的创建和使用方法。

8.2 习题参考答案

1. 选择题

(1) 存储过程是在 MySQL 服务器中定义并________的 SQL 语句集合。

A. 保存　　B. 执行　　C. 解释　　D. 编写

(2) 下面有关存储过程的叙述错误的是________。

A. MySQL 允许在存储过程创建时引用一个不存在的对象

B. 存储过程可以带多个输入参数，也可以带多个输出参数

C. 使用存储过程可以减少网络流量

D. 在一个存储过程中不可以调用其他存储过程

(3) MySQL 所支持的触发器不包括________。

A. insert 触发器　　B. delete 触发器

C. check 触发器　　D. update 触发器

(4) 下面有关触发器的叙述错误的是________。

A. 触发器是一个特殊的存储过程

B. 触发器不可以引用所在数据库以外的对象

C. 在一个表上可以定义多个触发器

D. 触发器在 check 约束之前执行

(5) MySQL 为每个触发器创建了两个临时表________。

A. max 和 min　　B. avg 和 sum　　C. int 和 char　　D. old 和 new

(6) 通过以下________语句临时关闭事件 e_test。

A. alter event e_test disable　　B. alter event e_test drop

C. alter event e_test enable　　D. alter event e_test delete

(7) 下列________语句用来定义游标。

A. create　　B. declare

C. declare...cursor for ...　　D. show

(8) 下列说法中错误的是________。

A. 常用触发器有 insert、update、delete 三种

B. 对于同一张数据表，可以同时有两个 before update 触发器

C. new 临时表在 insert 触发器中用来访问被插入的行

D. old 临时表中的值只读不能被更新

(9) 存储程序中的选择语句有________。

A. if　　B. while　　C. select　　D. switch

(10) 存储程序中不能使用的循环语句是________。

A. repeat　　B. while　　C. loop　　D. for

本题答案：B D C B D A C B A D

2. 思考题

(1) MySQL 中创建多条执行语句的存储过程或触发器时，为何总是遇到分号就结束创建，然后报错？如何解决这个问题？

答：MySQL 中，创建多条执行语句的触发器时，需要用到 begin...end 的形式。每个执行语句都必须以分号结束。但是，这样就会出问题。因为系统默认分号是 SQL 程序结束的

标志，遇到分号整个程序就结束了。要解决这个问题，就需要 delimiter 语句来改变程序的结束符号。如“delimiter &&”可以将程序的结束符号变成“&&”。如果要把结束符号变回分号，只要执行“delimiter ;”即可。

(2) 各种触发器的触发顺序是什么？

答：使用触发器时，触发器执行的顺序是 before 触发器、表操作(insert、update 和 delete)、after 触发器。

(3) 什么是事件？事件有什么作用？事件与触发器的区别有哪些？

答：事件就是需要在指定的时刻才被执行的某些特定任务，其中这些特定任务通常是一些确定的 SQL 语句。

事件可以根据需要在指定的时刻被事件调度器调用执行，并可以此取代原先只能由操作系统的计划任务来执行的工作。

事件和触发器相似，都是在某些事情发生的时候才被启动，但事件是基于特定时间周期触发来执行某些任务的，而触发器是基于某个表所产生的事件触发的，它们的区别也在于此。

(4) 简述游标在存储过程中的作用。

答：游标是一个被 select 语句检索出来的结果集。在存储了游标后，应用程序或用户就可以根据需要滚动或浏览其中的数据。

(5) 简述存储过程与存储函数的区别。

答：存储函数与存储过程之间存在的几点区别为：存储函数不能拥有输出参数，这是因为存储函数自身就是输出参数；而存储过程可以拥有输出参数。可以直接对存储函数进行调用，且不需要使用 call 语句；而对存储过程的调用，需要使用 call 语句。存储函数中必须包含一条 return 语句，而这条特殊的 SQL 语句不允许包含于存储过程中。

3. 上机练习题(以下题目默认数据库为 teaching)

(1) 创建存储过程 selectscore()，用指定的学号查询学生成绩。

答：代码和运行结果如下：

```
mysql> delimiter $$
mysql> create  procedure  selectscore(in s_no char(11))
    -> begin
    -> select *   from   score
    -> where studentno = s_no ;
    -> end  $$
      Query OK, 0 rows affected (0.01 sec)
mysql> delimiter ;
```

调用存储过程 selectscore()的代码和执行结果如下：

```
mysql> call selectscore('19112111208');
       +-----------+--------+-----+-------+
       | studentno   | courseno | daily |  final |
       +-----------+--------+-----+-------+
       | 19112111208 | c05109    |  76.5 |  91.0 |
       | 19112111208 | c06108    |  89.0 |  95.0 |
       +-----------+--------+-----+-------+
```

```
2 rows in set (0.08 sec)
Query OK, 0 rows affected (0.09 sec)
```

（2）编程在表 course 中创建一个触发器 course_detrigger，用于每次当删除表 course 中一行数据时，将会话变量 per1 的值设置为“old course deleted！”。

答：代码和运行结果如下：

```
mysql > create trigger course_detrigger after delete
    -> on course for each row set @per1 = 'old course deleted! ';
       Query OK, 0 rows affected (0.05 sec)
mysql > delete from course where courseno = 'c08171';
       Query OK, 1 row affected (0.20 sec)
mysql > select @per1;
        +----------------+
        | @per1                |
        +----------------+
        | old course deleted! |
        +----------------+
       1 row in set (0.00 sec)
```

（3）创建一个存储过程，用于实现给定表 student 中一个学生的姓名即可修改表 student 中该人的电子邮件地址为一个给定的值。

答：代码和运行结果如下：

```
mysql > delimiter $$
mysql > create procedure update_email(in user_name
    ->      char(11),in e_mail varchar(20))
    -> begin
    -> update student   set Email = e_mail where sname = user_name;
    -> end $$
       Query OK, 0 rows affected (0.04 sec)
mysql > delimiter ;
mysql > call update_email('封月明', 'feng@163.com');
       Query OK, 1 row affected (0.10 sec)
mysql > select studentno,sname,email from student where sname = '封月明';
        +-----------+-------+-------------+
        | studentno   | sname    | email           |
        +-----------+-------+-------------+
        | 19122203567 | 封月明   | feng@163.com |
        +-----------+-------+-------------+
       1 row in set (0.02 sec)
```

（4）创建一个事件，用于每 6 个月将表 score 中期末成绩高于 60 分的所有记录信息删除，该事件开始于下个月并且在 2019 年 12 月 31 日结束。

答：代码和运行结果如下：

```
mysql > delimiter $$
mysql > create event if not exists e_delete_final
    -> on schedule every 6 month
    -> starts curdate() + interval 2 month
```

```
    -> ends '2019-12-31'
    -> do
    -> begin
    -> if year(curdate())< 2019 then
    -> delete from score
    -> where final >= 60;
    -> end if;
    -> end   $$
       Query OK, 0 rows affected (0.09 sec)
mysql> delimiter ;
```

(5) 创建一个存储过程 scoreInfo，完成的功能是在表 student、表 course 和表 score 中查询以下字段：学号、姓名、性别、课程名称、期末分数。

答：代码和运行结果如下：

```
mysql> delimiter $$
mysql> create procedure scoreinfo()
    -> begin
    -> select student.studentno, sname, sex, cname, final
    -> from student, course, score
    -> where student.studentno = score.studentno
    ->          and score.courseno = course.courseno ;
    -> end   $$
       Query OK, 0 rows affected (0.03 sec)
```

(6) 创建一个带有参数的存储过程 stu_age，该存储过程根据输入的学号，在 student 表中计算此学生的年龄，并根据程序的执行结果返回不同的值，程序执行成功，返回整数 0，如果执行出错，则返回错误号。

答：代码和运行结果如下：

```
mysql> delimiter $$
mysql> create procedure stu_age(in stuno char(11))
    -> begin
    -> select (year(curdate()) - year(birthdate)) from student
    -> where studentno = stuno;
    -> set @er_val = @@error_count ;
    -> end $$
       Query OK, 0 rows affected (0.02 sec)
mysql> delimiter ;
mysql> call stu_age('19122203567');
        +---------------------------+
        | (year(curdate()) - year(birthdate)) |
        +---------------------------+
        |                        15 |
        +---------------------------+
       1 row in set (0.00 sec)
       Query OK, 0 rows affected (0.01 sec)
mysql> select @er_val;
        +--------+
        | @er_val |
```

```
+--------+
|       0 |
+--------+
1 row in set (0.00 sec)
```

(7) 创建事件 e_test，每天定时清空 test 表，5 天后停止执行。

答：代码和运行结果如下：

```
mysql> create event e_test
    -> on schedule every 1 day
    -> starts curdate()
    -> ends curdate() + interval 5 day
    -> do
    -> delete from  test ;
      Query OK, 0 rows affected (0.04 sec)
```

(8) 假设之前创建的 course 表没有设置外键级联策略，设置触发器，实现在 course 表中删除课程信息时，可自动删除该课程在 score 上的成绩信息。

答：代码和运行结果如下：

```
mysql> create trigger  detrigger  after delete
    -> on course  for each row
    -> delete from score
    -> where  score.courseno  = old.courseno ;
      Query OK, 0 rows affected (0.05 sec)
```

第9章 并发事务与锁机制

9.1 教学要求

9.1.1 基本要求

(1) 了解 MySQL 中的事务概念及事务运行机制,掌握事务 ACID 特性的内涵。

(2) 了解事务并发处理对数据的影响。

(3) 掌握事务的启用、删除、回滚、提交和设置检查点的常用操作。

(4) 掌握改变 MySQL 的自动提交模式的基本操作。

(5) 掌握事务隔离级别用途和设置事务的隔离级别的基本操作。

(6) 理解锁机制对于并发事务处理的作用。

(7) 掌握锁的分类方法和各种锁的应用场合。

(8) 了解死锁的概念和处理过程。

9.1.2 重点与难点

1. 本章重点

(1) 事务 ACID 特性的内涵。

(2) 事务并发处理对数据的影响。

(3) 事务的启用、删除、回滚、提交和设置检查点的常用操作。

(4) 改变 MySQL 的自动提交模式的基本操作。

(5) 设置事务的隔离级别的基本操作。

(6) 掌握锁的分类方法和各种锁的应用场合。

2. 本章难点

(1) MySQL 中的事务概念及事务运行机制。

(2) 事务并发处理对数据的影响。

(3) 事务的启用、删除、回滚、提交和设置检查点的常用操作。

(4) 理解锁机制对于并发事务处理的作用。

(5) 锁的分类方法和各种锁的应用场合。

(6) 死锁的概念和处理过程。

9.2　习题参考答案

1. 选择题

(1) MySQL 的事务不具有的特征是________。

A. 原子性　　B. 隔离性　　C. 一致性　　D. 共享性

(2) MySQL 中常见的锁类型不包括________。

A. 共享　　B. 意向　　C. 架构　　D. 排他

(3) 事务的隔离级别不包括________。

A. read uncommitted　　B. read committed

C. repeatable read　　D. repeatable only

(4) 死锁发生的原因是________。

A. 并发控制　　B. 服务器故障　　C. 数据错误　　D. 操作失误

(5) MySQL 中发生死锁需要________。

A. 用户处理　　B. 系统自动处理　　C. 修改数据源　　D. 取消事务

本题答案：D C D A B

2. 思考题

(1) 简述并发控制可能产生的影响，分别描述产生的原因。

答：多个用户访问同一个数据资源时，可能出现并发问题，其产生的原因包括如下 4 个方面：

① 更新丢失。当两个或多个事务选择同一行，然后根据最初选定的值更新该行时，就会出现更新丢失的问题。每个事务都不知道其他事务的存在。最后的更新将覆盖其他事务所做的更新，从而导致数据丢失。

② 不可重复读。当一个事务多次访问同一行且每次读取不同数据时，会出现不可重复读问题。因为其他事务可能正在更新该事务正在读取的数据。

③ 幻读。当对某行执行插入或删除操作，而该行属于某事务正在读取的行的范围时，就会出现幻读问题。

④ 脏读，即读出的是不正确的临时数据。例如，当第 2 个事务选择第 1 个事务正在更新的行时，就会出现此问题。第 2 个事务正在读取的数据尚未被其他事务提交，并可能由更新此行的事务更改。

(2) 如何设置事务的隔离级别？

答：事务可以设置隔离级别，隔离级别描述了一个事务必须与其他事务所进行的资源或数据更改相隔离的程度。隔离级别从允许并发负面影响(如脏读、幻读等)的角度进行描述。MySQL 支持的事务的隔离级别从最低到最高有如下 4 种：

- 未提交读(read uncommitted)；
- 已提交读(read committed)；
- 可重复读(repeatable read)；

• 可串行化(serializable)。

随着隔离级别的提高,可以更有效地防止数据的不一致性。但是,这将降低事务的并发处理能力,会影响多用户访问。隔离级别也可以通过编程方式进行设置。

(3) 如何在事务中设置保存点,保存点有什么用途?

答:在事务中保存点一般利用 save transaction 命令进行设置。设置保存点可以不用回滚到事务的起始位置,而是回滚到标记所在的位置即保存点,以提高事务执行的效率。

(4) 什么是死锁,哪些方法可以解除死锁?

答:死锁是指事务永远不会释放它们所占用的资源,死锁中的两个事务都将无限期等待。

MySQL 能够自动检测死锁循环,并选择一个会话作为死锁中放弃的一方,通过终止该事务来打断死锁。被终止的事务发生回滚,并返回给连接一个错误消息。

默认情况下:InnoDB 存储引擎一旦出现锁等待超时异常,InnoDB 存储引擎既不会提交事务,也不会回滚事务,而这是十分危险的。一旦发生锁等待超时异常,应用程序应该自定义错误处理程序,由程序开发人员选择进一步提交事务,还是回滚事务。

在 InnoDB 的事务管理和锁定机制中,有专门用于检测死锁的机制。当检测到死锁时,InnoDB 会选择产生死锁的两个事务中较小的一个产生回滚,而让另外一个较大的事务成功完成。需要注意的是,如果在产生死锁的场景中涉及不止 InnoDB 存储引擎时,InnoDB 是检测不到该死锁的,这时就只能通过锁定超时限制来解决该死锁了。

(5) 简述 MySQL 中锁的粒度及锁的常见类型。

锁的粒度是指锁的作用范围。锁的粒度可以分为服务器级锁(server-level locking)和存储引擎级锁(storage-engine-level locking)。MySQL 支持很多不同的表类型,而且对于不同的类型,锁定机制也是不同的。在 MySQL 中有三种锁定机制。

① 表锁:一个特殊类型的访问,整个表被客户锁定。根据锁定的类型,其他客户不能向表中插入记录,甚至从中读数据也受到限制。表级锁定的类型包括两种锁,即读锁(read)和写锁(write)。

② 行锁:行级的锁定比表级锁定或页级锁定对锁定过程提供了更精细的控制。在这种情况下,只有线程使用的行是被锁定的。表中的其他行对于其他线程都是可用的。行级锁定并不是由 MySQL 提供锁定机制,而是由存储引擎自己实现的,其中 InnoDB 的锁定机制就是行级锁定。行级锁定的类型包括三种:排他锁、共享锁和意向锁。

③ 页锁:MySQL 将锁定表中的某些行(称为页)。被锁定的行只对锁定最初的线程是可行的。

3. 上机练习题(本题利用 teaching 数据库中的表进行操作)

(1) 创建存储过程 up_score,实现在 score 表上执行 update 语句的事务,并执行存储过程进行验证。

答:代码和运行结果如下:

```
mysql> delimiter //
mysql> create procedure  up_score ()
    -> begin
    -> start transaction;
```

```
    -> update score
    -> set daily = 99,final = 99
    -> where studentno = '18125121107' and courseno = 'c05109';
    -> select * from  score
    -> where studentno = '18125121107' and courseno = 'c05109';
    -> commit ;
    -> end //
      Query OK, 0 rows affected (0.04 sec)
mysql> delimiter ;
mysql> call up_score ();
      +----------+--------+-----+----+
      | studentno    | courseno | daily | final |
      +----------+--------+-----+----+
      | 18125121107 | c05109    |  99.0 |  99.0 |
      +----------+--------+-----+----+
      1 row in set (0.04 sec)
      Query OK, 0 rows affected (0.34 sec)
```

(2) 定义一个事务,向 course 表中添加一条记录,并设置保存点。然后再删除该记录,并回滚到事务的保存点,提交事务。

答:代码和运行结果如下:

```
mysql> delimiter //
mysql> create procedure  insert_cno()
    -> begin
    -> start transaction;
    -> insert into course
    -> values('c05199','MySQL','必修',48,12,7);
    -> savepoint spcno_mysql;
    -> delete from course
    -> where courseno = 'c05199';
    -> rollback work  to  savepoint  spcno_mysql;
    -> select * from  course where courseno = 'c05199';
    -> commit ;
    -> end //
      Query OK, 0 rows affected (0.03 sec)
mysql> delimiter ;
mysql> call insert_cno();
      +-------+----+-----+-----+----+----+
      | courseno | cname | type | period | exp | term |
      +-------+----+-----+-----+----+----+
      | c05199   | MySQL | 必修 |    48 | 12 |      7 |
      +-------+----+-----+-----+----+----+
      1 row in set (0.03 sec)
      Query OK, 0 rows affected (0.08 sec)
```

(3) 创建事务,练习在 student 表上进行查询、插入和更新操作。

答:代码和运行结果如下:

```
mysql> delimiter //
mysql> create procedure  student_iud()
```

```
    -> begin
    -> start transaction;
    -> select studentno,sname
    ->   from student
    ->   where   studentno = '18135222201';
    ->   insert into student   values('1928261001','孙释远',
    ->      '女','1999 - 09 - 09',799,'1398909876','sdy@163.com');
    ->    update student   set sname = '孙释平'
    ->   where studentno = '19123567897';
    -> commit ;
    -> end //
      Query OK, 0 rows affected (0.02 sec)
mysql > delimiter ;
mysql > call student_iud();
       +-----------+-------+
       | studentno   | sname   |
       +-----------+-------+
       | 18135222201 | 凌浩风 |
       +-----------+-------+
      1 row in set (0.00 sec)
      Query OK, 0 rows affected (0.08 sec)
mysql > select studentno,sname,birthdate
    -> from   student where sname like '孙%';
       +-----------+-------+-----------+
       | studentno     | sname  | birthdate     |
       +-----------+-------+-----------+
       | 19123567897  | 孙释平 | 2002 - 08 - 04 |
       | 1928261001    | 孙释远 | 1999 - 09 - 09 |
       +-----------+-------+-----------+
      2 rows in set (0.01 sec)
```

(4) 将 course 表中课程号为 c08123 的课程名称改为“PHP 语言”,并提交该事务。

答:代码和运行结果如下:

```
mysql > delimiter //
mysql > create procedure   up_cname()
    -> begin
    -> start transaction;
    -> update course set cname = 'PHP 语言'
    -> where courseno = 'c08123';
    -> commit;
    -> select * from   course where courseno = 'c08123';
    -> end //
      Query OK, 0 rows affected (0.01 sec)
mysql > delimiter ;
mysql > call up_cname();
       +-------+-------+-----+------+---+----+
       | courseno | cname     | type | period | exp | term |
       +-------+-------+-----+------+---+----+
       | c08123    | PHP 语言 | 必修 |     40 |   0 |   5 |
       +-------+-------+-----+------+---+----+
      1 row in set (0.03 sec)
      Query OK, 0 rows affected (0.06 sec)
```

第10章　权限管理及安全控制

10.1　教学要求

10.1.1　基本要求

(1) 了解 MySQL 权限系统的工作机制,了解 MySQL 数据库中各表的功能。
(2) 掌握 MySQL 权限系统的工作过程。
(3) 掌握普通用户的创建、使用和删除的命令。
(4) 掌握 MySQL 命令的使用格式和执行过程。
(5) 了解利用图形工具管理用户的过程。
(6) 了解 MySQL 的权限类型。
(7) 掌握授权管理、收回、查看和限制权限的基本操作。
(8) 了解 MySQL 数据库安全的常见问题。
(9) 了解确保 MySQL 安全的注意事项。

10.1.2　重点与难点

1. 本章重点

(1) MySQL 权限系统的工作过程。
(2) 普通用户的创建、使用和删除。
(3) 掌握 MySQL 命令的使用格式和执行过程。
(4) 掌握管理访问权限的 grant 和 revoke 命令格式、应用方法和主要用途。

2. 本章难点

(1) MySQL 权限系统的工作过程。
(2) 普通用户的创建、使用和删除。
(3) MySQL 命令的使用格式和执行过程。
(4) 授权管理、收回、查看和限制权限的基本操作。
(5) MySQL 数据库安全常见问题。

10.2　习题参考答案

1. 选择题

(1) 在 MySQL 中,可以使用________语句来为指定数据库添加用户。

A. revoke　　B. grant　　C. insert　　D. create

(2) MySQL 中存储用户全局权限的表是________。

A. tables_priv　　B. procs_priv　　C. columns_priv　　D. user

(3) 下列命令中,________命令用于撤销 MySQL 用户对象权限。

A. revoke　　B. grant　　C. deny　　D. create

(4) 给名字是 liping 的用户分配对数据库 teaching 中的 student 表的查询和插入数据权限的语句是________。

A. grant select,insert on teaching.student for 'liping'@'localhost'

B. grant select,insert on teaching.student to 'liping'@'localhost'

C. grant 'liping'@'localhost' to select,insert for teaching.student

D. grant 'liping'@'localhost' to teaching.student on select,insert

(5) 修改自己的 MySQL 服务器密码的命令是________。

A. mysql　　B. grant

C. set password　　D. change password

本题答案:C D A B C

2. 思考题

(1) 简述 MySQL 在对象上进行权限设置时,用户和权限的关系。

答:对象权限表示对特定的数据库对象(表、视图、字段和存储过程)的操作权限,它决定了能对表、视图等数据库对象执行的操作。在对象上进行权限设置时,用户可以单独设置权限,但需要防止权限设置冲突。

(2) 在 MySQL 中可以授予的权限有哪几个层次?

答:在 MySQL 中可以授予的权限有如下权限组:列权限、表权限、数据库权限和用户权限。

(3) 在 MySQL 的权限授予语句中,可用于指定权限级别的值有哪几类格式?

答:在 MySQL 的权限授予语句中,可用于指定权限级别的值有下面几类格式。

- *:表示当前数据库中的所有表。
- *.*:表示所有数据库中的所有表。
- db_name.*:表示某个数据库中的所有表。
- db_name.tbl_name:表示某个数据库中的某个表或视图,db_name 指定数据库名,tbl_name 指定表名或视图名。
- table_name:表示某个表或视图,table_name 指定表名或视图名。
- db_name.routine_name:表示某个数据库中的某个存储过程或函数,routine_name 指定存储过程名或函数名。

(4) 在数据加密过程中,password()函数有什么作用?

答:当使用 set password、insert 或者 update 指定账户的密码时,必须用 password()函数对它进行加密。因为 user 表以加密方式保存密码,而不是明文。当用户使用该密码连接服务器时,连接使用的密码值将被加密,并同保存在 user 表中的密码进行比较。因此比较

将失败，服务器会拒绝连接。唯一的特例是如果密码为空，则不需要使用 password()。

3. 上机练习题(本题利用 teaching 数据库进行操作)

(1) 使用 grant 语句创建一个新用户 ex_user，密码为 pass123。用户 ex_user 对当前数据库中的所有表有查询、插入权限，并授予 grant 权限。

答：代码和运行结果如下：

```
mysql > grant select,insert on * to 'ex_user'@'localhost'
    -> identified by 'pass123'
    -> with grant option;
      Query OK, 0 rows affected, 1 warning (0.22 sec)
```

(2) 利用 grant 语句将 teaching 数据库中 student 表的 delete 权限授予用户 ex_user。

答：代码和运行结果如下：

```
mysql > grant delete on teaching.student
    -> to 'ex_user'@'localhost';
      Query OK, 0 rows affected (0.09 sec)
```

(3) 收回 ex_user 用户对 teaching 数据库中 student 表的 delete 权限。

答：代码和运行结果如下：

```
mysql > revoke delete on teaching.student
    -> from ex_user@localhost;
      Query OK, 0 rows affected (0.02 sec)
```

(4) 使用 revoke 语句收回 ex_user 用户的所有权限，包括 grant 权限。

答：代码和运行结果如下：

```
mysql > revoke all privileges,grant option
    -> from ex_user@localhost;
      Query OK, 0 rows affected (0.02 sec)
```

(5) 假定当前系统中不存在用户 swming，请编写一段 SQL 语句，要求创建这个新用户，并为其设置对应的系统登录口令"my123"，同时授予该用户在数据库 teaching 的表 course 中拥有 select 和 update 的权限。

答：代码和运行结果如下：

```
mysql > grant select, update on teaching.course to 'swming'@'localhost'
    -> identified  by 'my123';
      Query OK, 0 rows affected, 1 warning (0.04 sec)
```

第11章 备份与恢复

11.1 教学要求

11.1.1 基本要求

（1）了解数据库备份的原因和分类标准。

（2）了解数据库恢复时需要注意的时机问题。

（3）掌握数据库备份的基本格式和基本操作。

（4）掌握数据库还原的基本格式和基本操作。

（5）掌握表数据的导入与导出基本操作。

（6）了解数据库迁移的各种方法。

11.1.2 重点与难点

1. 本章重点

（1）使用 mysqldump 命令备份。

（2）直接复制整个数据库目录。

（3）使用 mysqlhotcopy 工具快速备份。

（4）利用 MySQL 命令恢复数据。

（5）用 source 恢复表和数据库。

（6）用 select...into outfile 导出文件。

（7）用 MySQL 命令导出文本文件。

（8）用 load data infile 方式导入文本文件。

2. 本章难点

（1）直接复制整个数据库目录。

（2）使用 mysqlhotcopy 工具快速备份。

（3）数据库迁移的各种方法。

（4）用 select...into outfile 导出文件。

（5）用 load data infile 方式导入文本文件。

11.2 习题参考答案

1. 选择题

(1) 在数据库系统生命周期中可能发生的灾难不包括________。

A. 系统故障　B. 事务故障　C. 掉电故障　D. 介质故障

(2) 按备份时服务器是否在线划分不包括________备份。

A. 热备份　B. 完全备份　C. 冷备份　D. 温备份

(3) 还原数据库时，首先要进行________操作。

A. 创建数据表备份　B. 创建完整数据库备份

C. 创建冷设备　D. 删除最近事务日志备份

(4) 创建数据库文件或文件组备份时，首先要进行________操作。

A. 创建事务日志　B. 创建完整数据库备份

C. 创建温备份　D. 删除差异备份

(5) 下面故障发生时，________需要数据库管理员进行手工操作恢复。

A. 停电　B. 误删表数据　C. 死锁　D. 操作系统错误

本题答案：C B B B B

2. 思考题

(1) 为什么在 MySQL 中需要进行数据库的备份与恢复操作？

答：备份和恢复的目的就是将数据库中的数据进行导出，生成副本，然后在系统发生故障后能够恢复全部或部分数据。数据备份就是制作数据库结构、对象和数据的拷贝，以便在数据库遭到破坏时，或因需求改变而需要把数据还原到改变以前时能够恢复数据库。数据恢复就是指将数据库备份加载到系统中。数据备份和恢复可以用于保护数据库的关键数据。在系统发生错误或者因需求改变时，利用备份的数据可以恢复数据库中的数据。

(2) MySQL 数据库备份与恢复的常用方法有哪些？

答：MySQL 数据库备份与恢复的常用方法有：

- 使用 SQL 语句备份和恢复表数据。
- 使用 MySQL 客户端实用程序备份和恢复数据。
- 使用 MySQL 图形界面工具备份和恢复数据。
- 直接复制。

(3) 使用直接复制方法实现数据库备份与恢复时，需要注意哪些事项？

答：使用直接从一个 MySQL 服务器复制文件到另一个服务器的方法，需要特别注意以下两点：

① 两个服务器必须使用相同或兼容的 MySQL 版本。

② 两个服务器必须硬件结构相同或相似，除非要复制的表使用 MyISAM 存储格式，这是因为这种表可以为在不同的硬件体系中共享数据提供保证。

(4) 进行数据库还原应该注意哪些问题?

答:在还原数据库的时候,应该注意如下问题:

① 了解执行的备份方法的类型和备份是否存在。

② 确认备份文件包含要还原的备份,并确认备份是否有效。

③ 如果使用完整数据库备份来还原数据库,MySQL 重新创建这些数据库文件和所有的数据库对象,如果使用差异数据库备份来恢复,则可以恢复最近的差异数据库备份。

④ 在数据库还原过程中确保要还原的数据库没有打开的连接。因为在进行还原的时候不允许有连接到数据库的连接。

(5) 备份数据库的时机如何选择?

答:备份数据库的时机取决于可接受的数据丢失量和数据库活动的频繁程度。可以从下列几方面考虑备份的时机。

- 创建数据库或为数据库填充了数据以后,用户应该备份数据库。
- 创建索引后备份数据库。
- 清理事务日志后备份数据库。当执行了清理事务日志的语句后,应该备份数据库。在清理之后,事务日志将不包含数据库的活动记录,也不能用来还原数据库。
- 执行了无日志操作后也应该备份数据库。

3. 上机练习题(本题利用 teaching 数据库进行操作)

(1) 使用 mysqldump 命令备份数据库 teaching 中的所有表。

答:代码和运行结果如下:

```
mysqldump -u root -p teaching > d:/bak/teachingbak.sql
Enter password: ******
```

(2) 使用 source 命令将备份文件 teachingbak.sql 恢复到数据库中。

答:代码和运行结果如下:

```
mysql > source d:\bak\teachingbak.sql
      Query OK, 0 rows affected (0.00 sec)
      ...
      Query OK, 0 rows affected (0.00 sec)
```

(3) 使用 mysqldump 命令备份数据库中的 score 表。

答:代码和运行结果如下:

```
mysqldump -u root -p teaching course > d:/bak/score.sql
Enter password: ******
```

(4) 删除 score 表的数据,用 source 命令恢复。

答:代码和运行结果如下:

```
#尝试删除 score 表的数据
mysql > use teaching;
Database changed
mysql > delete  from score;
      Query OK, 26 rows affected (0.11 sec)
#利用放在"d:/bak"路径下 score 的备份文件 score.sql,使用 source 命令把备份好的文件导入进行恢复
```

```
mysql> source d:/bak/score.sql;
       Query OK, 0 rows affected (0.00 sec)
       ...
       Query OK,26 rows affected (0.03 sec)
       Records:26 Duplicates: 0   Warnings: 0
       ...
       Query OK, 0 rows affected (0.00 sec)
```

(5) 使用 MySQL 命令将 teaching 数据库中的 course 表中的记录导出到文本文件。

答：代码和运行结果如下：

```
mysql -uroot -p --execute="select * from course;"teaching>d:/bak/teach.txt
Enter password: ******
```

(6) 用备份好的 teach. txt 文件恢复 course 表数据。为避免主键冲突，要用 replace into table 直接将数据进行替换来恢复数据。

答：代码和运行结果如下：

```
mysql> load data infile 'd:/bak/teach.txt'
    -> replace into  table course ;
       Query OK,13 rows affected (0.03 sec)
       Records:26 Duplicates: 0   Warnings: 0
```

第 12 章 MySQL 性能优化

12.1 教学要求

12.1.1 基本要求

(1) 了解数据库优化的含义和查看数据库性能参数的方法。
(2) 掌握优化查询的方法、优化数据库结构的方法和优化 MySQL 服务器的方法。
(3) 了解使用索引优化查询的方法。
(4) 了解在 MySQL 中分析查询效率的方法。
(5) 掌握在 MySQL 中应用高速缓存提高查询性能的方法。
(6) 掌握如何在多表查询中提高查询性能。
(7) 掌握在 MySQL 中使用临时表提高优化查询效率的方法。

12.1.2 重点与难点

1. 本章重点

(1) MySQL 性能优化的主要方法。
(2) 优化查询语句的使用方法。
(3) 索引对查询速度的影响。
(4) 使用索引优化查询的方法。
(5) 优化数据库结构的方法。
(6) 优化慢查询的方法。
(7) 查询高速缓存的过程。

2. 本章难点

(1) 优化查询语句的使用方法。
(2) 索引对查询速度的影响。
(3) 使用索引优化查询的方法。
(4) 优化数据库结构的方法。
(5) 查询高速缓存的过程。

12.2 习题参考答案

1. 选择题

(1) 使用 Explain 语句可对________语句的执行效果进行分析,通过分析提出优化运行速度的方法。

A. select B. insert C. delete D. create

(2) 多列索引在表的多个字段上创建一个索引。只有查询条件中使用了这些字段中的________时,索引才会被正常使用。

A. 最后 1 个字段 B. 第 2 个字段 C. 第 1 个字段 D. 所有字段

(3) 使用 analyze table 分析表的过程中,数据库系统会对表加一个________。在分析期间,只能读取表中的记录,不能更新和插入记录。

A. 排他锁 B. 只读锁 C. 读写锁 D. 意向锁

(4) 若有某些查询经常涉及多表连接,可以视情况将这些字段建立一个________来进行查询和统计,提高查询效率。

A. 查询表 B. 排序表 C. 中间表 D. 子查询

(5) 为了解决插入记录时,________过程会降低插入记录速度的情况,在插入记录之前可以先禁用索引。等到记录都插入完毕后再开启索引。

A. 索引 B. 排序 C. 查询 D. 插入

本题答案：A C B C A

2. 思考题

(1) 如何使用查询缓存区?

答：Query_cache_type：表示查询缓冲区的开启状态。0 表示关闭,1 表示开启。查询缓冲区主要是为了提高经常执行相同的查询操作的速度,但是,另一方面查询缓冲区也无形中增加了系统的开销,所以有时为减少系统的开销,也可以关闭查询缓冲区。如果设置 Query_cache_type=1,所有查询都将使用查询缓冲区。

如果设置 Query_cache_type=0,则所有查询不使用查询缓冲区。但不会导致 MySQL 释放 query_cache_size 配置的缓冲区内存。如果要禁用查询缓存,可以设置 query_cache_size=0,禁用查询缓存后,将没有明显的开销。

(2) 为什么查询语句中的索引有时会没有发挥作用?

答：在查询过程中使用索引,势必会提高数据库的查询效率,应用索引来查询数据库中的内容,可以减少查询的记录数,从而达到查询优化的目的。

一般情况下,使用索引可以提高查询的速度,但如果 MySQL 语句使用不恰当的话,索引将无法发挥它应有的作用。如果在一个表中创建了多列的复合索引,只有查询条件中使用了这些字段的第一个字段时,索引才会使用。

如果一个查询,没有使用索引进行查询,则使用 explain 命令分析查询时,type 为 all 表示要对表进行全扫描。

(3) 什么是慢查询？形成慢查询的原因有哪些？

答：MySQL 将执行时间长的查询语句称为慢查询。MySQL 利用 long_query_time 来定义慢于多少秒的才算“慢查询”，系统默认是 10 秒。并可以通过设置 slow_query_log 参数决定是否打开日志记录慢查询，on 表示打开，off 表示关闭。

形成慢查询的原因可以从如下几个方面去了解：

① 了解业务方使用场景，查看查询字段是否没有索引或者没有用到索引(这是查询慢最常见的问题，是程序设计的缺陷)。

② 硬件环境问题。内存不足，网络速度慢，I/O 吞吐量小，形成了瓶颈效应。

③ 有没有创建计算列导致查询不优化；查询语句不好，没有优化，是否返回了不必要的数据；查询出的数据量过大(可以采用多次查询，其他的方法降低数据量)。

④ 利用 explain 查看执行计划，是否与预期一致(从锁定记录较少的表开始查询)。

⑤ 是否形成锁或者死锁(这也是查询慢最常见的问题，是程序设计的缺陷)。

⑥ 利用 sp_lock、sp_who 等参数查看活动的用户，是否存在读写竞争资源。

(4) 如何从优化查询的角度进行表字段的设计？

答：在设计数据表时应优先考虑使用特定字段长度，后考虑使用变长字段，如在用户创建数据表时，考虑创建某个字段类型为 varchar 而设置其字段长度为 255，但是在实际应用时，该用户所存储的数据根本达不到该字段所设置的最大长度，命令外如设置用户性别的字段，往往可以用 M 表示男性，F 表示女性，如果给该字段设置长度为 varchar(50)，则该字段占用了过多列宽，这样不仅浪费资源，也会降低数据表的查询效率。适当调整列宽不仅可以减少磁盘空间，同时也可以使数据在进行处理时产生的 I/O 过程减少。将字段长度设置成其可能应用的最大范围可以充分地优化查询效率。

3. 上机练习题(本题利用 teaching 数据库中的表进行操作)

(1) 使用 explain 语句来分析一个查询语句。

答：代码和运行结果如下：

```
mysql> explain  select  *  from  student\G
      ************************ 1. row ***************************
             id: 1
    select_type: SIMPLE
          table: student
     partitions: NULL
           type: ALL
  possible_keys: NULL
            key: NULL
        key_len: NULL
            ref: NULL
           rows: 11
       filtered: 100.00
          Extra: NULL
1 row in set, 1 warning (0.00 sec)
```

(2) 分析查询语句，对比不使用索引和使用索引的情况。

答：代码和运行结果如下：

```
#分析未使用索引时的查询情况
mysql> explain select * from course where cname = '软件工程'\G
        ******************** 1. row ********************
                   id: 1
          select_type: SIMPLE
                table: course
           partitions: NULL
                 type: ALL
        possible_keys: NULL
                  key: NULL
              key_len: NULL
                  ref: NULL
                 rows: 13
             filtered: 10.00
                Extra: Using where
        1 row in set, 1 warning (0.00 sec)
mysql> create  index  index_cname  on  course(cname); -- 创建索引
        Query OK, 0 rows affected, 1 warning (0.56 sec)
        Records: 0  Duplicates: 0  Warnings: 1
#分析使用索引时的查询情况
mysql> explain select * from course where cname = '软件工程'\G
        ********************** 1. row **********************
                   id: 1
          select_type: SIMPLE
                table: course
           partitions: NULL
                 type: ref
        possible_keys: index_cname
                  key: index_cname
              key_len: 12
                  ref: const
                 rows: 1
             filtered: 100.00
                Extra: NULL
        1 row in set, 1 warning (0.00 sec)
```

(3) 利用 explain 语句执行查询命令，应用 like 关键字，且匹配字符串中含有的百分号“%”。

答：代码和运行结果如下：

```
mysql> explain select * from course where cname like '%软件'\G
        ******************* 1. row *******************
                   id: 1
          select_type: SIMPLE
                table: course
           partitions: NULL
                 type: ALL
        possible_keys: NULL
                  key: NULL
              key_len: NULL
                  ref: NULL
                 rows: 13
             filtered: 11.11
```

```
           Extra: Using where
     1 row in set, 1 warning (0.00 sec)
mysql> explain select * from course where  cname like '软件%'\G
     ******************* 1. row *******************
              id: 1
     select_type: SIMPLE
           table: course
      partitions: NULL
            type: range
   possible_keys: index_cname
             key: index_cname
         key_len: 12
             ref: NULL
            rows: 1
        filtered: 100.00
           Extra: Using index condition
     1 row in set, 1 warning (0.00 sec)
```

说明：

从上面两个运行结果中可以看出，cname 列使用了索引，都与 like 关键字进行匹配。如果匹配字符（%或_）在字符串的后面，索引在其中起作用，如第一种情况，匹配字符（%或_）在字符串的前面，索引将不起作用，type 值为 all 级，即对表进行全扫描，检查的行数为 13 行，虽然符合条件的记录行为 1 行。

对于第二种情况，type 值为 range 级，因为有 1 条符合条件的记录，所以 rows 参数值为 1，检查的行数只有 1 行。

由此可以知道，使用 like 关键字和通配符的做法虽然简单、易懂，但却也是以牺牲系统性能为代价的。

(4) 执行 analyze table 语句分析 course 表。

答：代码和运行结果如下：

```
mysql> analyze table  course;
```

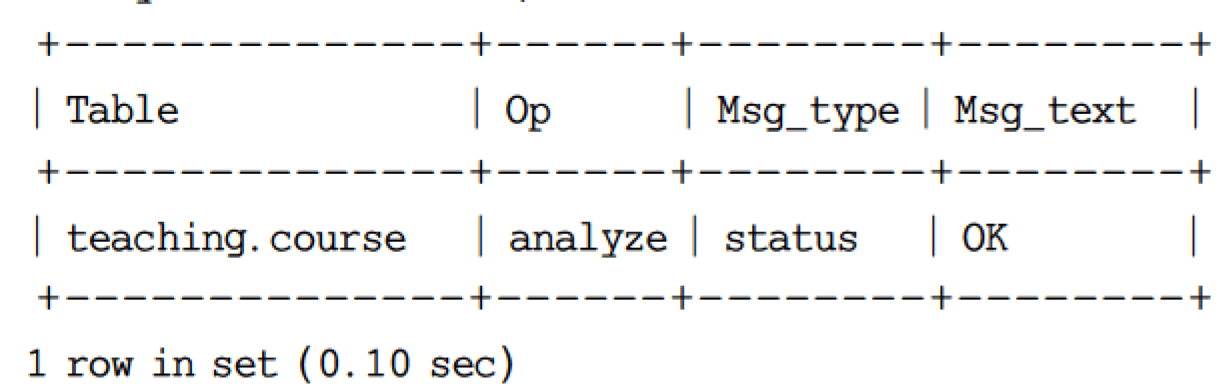


Table	Op	Msg_type	Msg_text
teaching.course	analyze	status	OK

```
     1 row in set (0.10 sec)
```

(5) 执行 check table 语句检查表 course。

答：代码和运行结果如下：

```
mysql> check table  course;
```

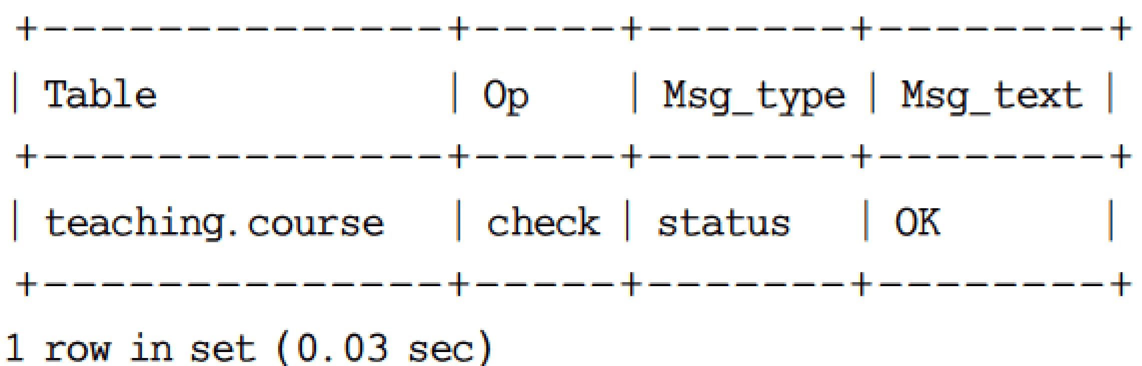


Table	Op	Msg_type	Msg_text
teaching.course	check	status	OK

```
     1 row in set (0.03 sec)
```

第13章　MySQL 日志文件管理

13.1　教学要求

13.1.1　基本要求

(1) 了解 MySQL 日志文件的分类和作用。
(2) 掌握启用、设置、查看和删除错误日志的基本操作。
(3) 掌握启用、查看、清理和暂时停止二进制日志的基本操作。
(4) 掌握利用二进制日志恢复数据库的操作。
(5) 了解通用查询日志的管理。
(6) 掌握启用、操作和删除慢查询日志的基本操作。

13.1.2　重点与难点

1. 本章重点

(1) 二进制日志的启用、查看、暂停和清理的基本操作。
(2) 利用二进制日志恢复数据库的过程。
(3) 错误日志的启用、查看和删除等基本操作。
(4) 通用查询日志的启用、查看和删除等基本操作。

2. 本章难点

(1) 错误日志的启用、查看和删除等基本操作。
(2) 利用二进制日志恢复数据库的过程。
(3) 通用查询日志的启用、查看和删除等基本操作。
(4) 设置错误日志存取路径的方法。

13.2　习题参考答案

1. 选择题

(1) MySQL 的日志在默认情况下，只启动了________的功能。
A. 二进制日志　　B. 错误日志　　C. 通用查询日志　　D. 慢查询日志
(2) MySQL 的日志中，除________外，其他日志都是文本文件。
A. 二进制日志　　B. 错误日志　　C. 通用查询日志　　D. 慢查询日志
(3) 如果很长时间不清理二进制日志，将会浪费很多的磁盘空间。删除二进制日志的

方法不包括________。

A. 删除所有二进制日志　　B. 删除指定编号的二进制日志

C. 根据创建时间来删除二进制日志　　D. 删除指定时刻的二进制日志

(4) 如果数据库遭到意外损坏,首先应该使用最近的备份文件来还原数据库,可以使用________来还原。

A. 通用查询日志　B. 错误日志　C. 二进制日志　D. 慢查询日志

本题答案:B A D C

2. 思考题

(1) MySQL 的日志分几类,各有什么作用?

答:包括二进制日志、错误日志、通用查询日志和慢查询日志 4 类。4 类日志文件的具体功能如下:

- 二进制日志:以二进制文件的形式记录了数据库中所有更改数据的语句,还可以运用于复制操作。
- 错误日志:记录 MySQL 服务的启动、运行和停止 mysqld 时出现的问题。
- 通用查询日志:记录用户登录和记录查询的信息。
- 慢查询日志:记录所有执行时间超过 long_query_time 秒的查询或不使用索引的查询。

(2) 慢查询日志有什么特点和作用?

答:慢查询日志的特点:慢查询日志是记录查询时长超过指定时间的日志。执行时间超过指定时间的查询语句会被记录到慢查询日志中。如果查询时间超过了这个时间值,这个查询语句将被记录到慢查询日志。默认情况下,慢查询日志功能是关闭的。

慢查询日志的作用:慢查询日志主要用来记录执行时间较长的查询语句。通过慢查询日志,可以找出执行时间较长、执行效率较低的语句,然后进行优化。如果用户希望查询哪些查询语句的执行效率低,可以从慢查询日志中获得想要的信息。慢查询日志也是以文本文件的形式存储的。可以使用普通的文本文件查看工具来查看。

(3) 简述 MySQL 日志的主要作用。

答:MySQL 日志用来记录 MySQL 数据库的运行情况、用户操作和错误信息等。当 MySQL 服务器在某个时间出现异常时,异常信息也会被记录到日志文件中。日志文件可以为 MySQL 管理和优化提供必要的信息。

如果 MySQL 数据库系统意外停止服务,可以通过错误日志查看出现错误的原因。并且可以通过二进制日志文件来查看用户执行了哪些操作,对数据库文件做了哪些修改等。然后根据二进制日志文件的记录来修复数据库。

需要了解的是,启动日志功能会降低 MySQL 数据库的性能。同时,日志会占用大量的磁盘空间。如果是用户量非常大,操作又非常频繁的数据库,日志文件需要的存储空间甚至比数据库文件需要的存储空间还要大。

3. 上机练习题(本题利用 teaching 数据库进行操作)

(1) 使用 show variables 语句查询当前日志设置。

答：代码和运行结果如下：

```
mysql> show  variables  like  'log_%';
      +----------------------------+-------------------+
      | Variable_name              | Value             |
      +----------------------------+-------------------+
      | log_bin                    | ON                |
      | log_bin_basename          | C:\ProgramData\MySQL\MySQL Server
                                    5.7\Data\PGIG1MIWMYPOFBS-bin   |
      | log_bin_index            | C:\ProgramData\MySQL\MySQL Server
                                 5.7\Data\PGIG1MIWMYPOFBS-bin.index|
      | log_bin_trust_function_creators | OFF          |
      | log_bin_use_v1_row_events  | OFF               |
      | log_builtin_as_identified_by_password | OFF    |
      | log_error                | .\PGIG1MIWMYPOFBS.err |
      | log_error_verbosity        | 3                 |
      | log_output                 | FILE              |
      | log_queries_not_using_indexes | OFF            |
      | log_slave_updates          | OFF               |
      | log_slow_admin_statements  | OFF               |
      | log_slow_slave_statements  | OFF               |
      | log_statements_unsafe_for_binlog | ON          |
      | log_syslog                 | ON                |
      | log_syslog_tag             |                   |
      | log_throttle_queries_not_using_indexes |  0    |
      | log_timestamps             | UTC               |
      | log_warnings               | 2                 |
      +----------------------------+-------------------+
      19 rows in set, 1 warning (0.00 sec)
```

(2) 使用 show binary logs 查看二进制日志文件的个数及文件名。

答：代码和运行结果如下：

```
mysql> show  binary  logs;
      +----------------------+---------+
      | Log_name             |File_size |
      +----------------------+---------+
      | PGIG1MIWMYPOFBS-bin.000037 |  177 |
      | PGIG1MIWMYPOFBS-bin.000038 |  699 |
      | PGIG1MIWMYPOFBS-bin.000039 | 9799 |
      | PGIG1MIWMYPOFBS-bin.000040 | 4268 |
      | PGIG1MIWMYPOFBS-bin.000041 | 9456 |
      | PGIG1MIWMYPOFBS-bin.000042 | 2562 |
      +----------------------+---------+
      6 rows in set (0.06 sec)
```

(3) 使用 purge master logs 删除 2017 年 8 月 30 日前创建的所有日志文件。

答：代码和运行结果如下：

```
mysql> purge master logs before '20170830';
      Query OK, 0 rows affected, 1 warning (0.11 sec)
```

```
    #语句执行之后,2017 年 8 月 30 日之前创建的日志文件都将被删除,但 2017 年 8 月 30 日的
#日志会被保留
```

(4) 使用记事本查看 MySQL 错误日志。

答：代码和运行结果如下：

```
#通过 show variables 语句查询错误日志的存储路径和文件名
  show variables like 'log_error';
#可以看到错误的文件是 log_error,位于 MySQL 默认的数据目录下,使用记事本打开该文件,可以看
#到 MySQL 的错误日志
  2017-06-30T09:53:27.921504Z 0 [Note] InnoDB: Number of pools: 1
  2017-06-30T09:53:28.046304Z 0 [Note] InnoDB: Not using CPU crc32 instructions
  2017-06-30T09:53:32.772313Z 0 [Note] InnoDB: Waiting for purge to start
  2017-07-23T13:42:31.071708Z 0 [Note] Shutting down plugin 'INNODB_CMPMEM'
  2017-07-23T13:42:31.071708Z 0 [Note] Shutting down plugin 'INNODB_CMP_RESET'
  2017-07-23T13:42:31.071708Z 0 [Note] Shutting down plugin 'INNODB_CMP'
  2017-07-23T13:42:31.071708Z 0 [Note] Shutting down plugin 'INNODB_LOCKS'
  2017-07-23T13:42:31.071708Z 0 [Note] Shutting down plugin 'INNODB_TRX'
  2017-07-23T13:42:31.071708Z 0 [Note] Shutting down plugin 'InnoDB'
  2017-07-25T23:49:09.261507Z 0 [Note] End of list of non-natively partitioned tables
  2017-07-25T23:49:09.261507Z 0 [Note] C:\Program Files\MySQL\MySQL Server 5.7\bin\
mysqld.exe: ready for connections.
  Version: '5.7.17-log'  socket: ''port: 3306  MySQL Community Server (GPL)
```

以上是错误日志文件的一小部分,这里面记载了系统的一些错误。

第14章　使用 PHP 操作 MySQL 数据库

14.1　教学要求

14.1.1　基本要求

(1) 了解 PHP 语言的特点和工作原理。
(2) 掌握搭建 PHP+MySQL 的集成开发环境的过程。
(3) 了解配置集成开发环境(IDE)常用软件 NetBeans 的安装和配置过程。
(4) 学会安装和配置 Apache 软件。
(5) 掌握 PHP 语言连接 MySQL 服务器的操作步骤。
(6) 掌握使用 PHP 管理 MySQL 数据库的基本操作语句。
(7) 掌握使用 PHP 处理 MySQL 结果集的语句。
(8) 了解使用 mysqli_free_result()函数释放内存的过程。

14.1.2　重点与难点

1. 本章重点

(1) 搭建 PHP+MySQL 的集成开发环境的过程。
(2) 配置集成开发环境(IDE)常用软件 NetBeans 的安装和配置过程。
(3) 安装和配置 Apache 软件。
(4) 使用 PHP 管理 MySQL 数据库的基本操作语句。
(5) 使用 PHP 处理 MySQL 结果集的语句。

2. 本章难点

(1) PHP 语言的工作原理。
(2) 搭建 PHP+MySQL 的集成开发环境的过程。
(3) 安装和配置 Apache 软件。
(4) PHP 语言连接 MySQL 服务器的操作。
(5) 使用 PHP 管理 MySQL 数据库的基本操作。
(6) 使用 PHP 处理 MySQL 结果集的语句。

14.2 习题参考答案

1. 思考题

(1) 简述 PHP 语言的基本特点。

答：PHP 是英文超级文本预处理语言 Hypertext Preprocessor 的缩写。PHP 是一种 HTML 内嵌式的语言，是一种在服务器端执行的嵌入 HTML 文档的脚本语言，语言的风格类似于 C 语言，被广泛运用。

(2) 简述利用 PHP 与 MySQL 数据库连接的步骤。

答：使用 PHP 进行 MySQL 数据库编程的基本步骤如下：

① 建立与 MySQL 数据库服务器的连接。

② 选择要对其进行操作的数据库。

③ 再执行相应的数据库操作，包括对数据的添加、删除、修改和查询等。

④ 关闭与 MySQL 数据库服务器的连接。

(3) 设置结果集数组的表述方式中的 mysqli_assoc、mysqli_num 和 mysqli_both 分别表示什么？

答：使用 mysqli_fetch_array()函数将结果集返回到数组的操作过程中，使用 mysqli_query()函数执行 select 语句时，将成功返回查询结果集，返回结果集后，使用 mysqli_fetch_array()函数可以获取查询结果集信息，并放入到一个数组中，这些数组的类型如下：

- mysqli_assoc：表示数组采用关联索引。
- mysqli_num：表示数组采用数字索引。
- mysqli_both：同时包含关联和数字索引的数组。

其中，mysqli_both 是默认值。

(4) 简述 mysqli_query()函数的作用。

答：使用 mysqli_query()函数可以执行 SQL 语句。在 PHP 中，通常使用 mysqli_query()函数来执行对数据库操作的 SQL 语句，包括对数据进行查询、插入、更新和删除等操作。mysqli_query()函数一次只能执行一条 SQL 语句。如果 SQL 语句是 insert 语句、update 语句、delete 语句等，语句执行成功，mysqli_query()函数返回 true，否则返回 false。并且可以通过 mysqli_affected_rows()函数获取发生变化的记录数。

2. 上机练习题(本题利用 teaching 数据库中的表进行操作)

(1) 利用 PHP 语言查询数据表 course 中的数据。

答：代码和运行结果如下：

```
<?php
# 连接 mysql 数据库服务器
$ conn1 = mysqli_connect("localhost","root","123456");
$ select = mysqli_select_db( $ conn1,"teaching");          //连接服务器中的 teaching
if( $ select)
    {
    header("Content - Type:text/html;charset = gb2312");    //设置字符集
  echo "数据库连接成功!" ;                                   //判断是否连接成功
```

```
    }
$ query = "select * from  course";
$ result = mysqli_query( $ conn1, $ query) or die("查询失败!".mysqli_error());
echo mysqli_affected_rows( $ conn1);
?>
```

运行结果：

```
数据库连接成功!13
```

(2) 利用 mysqli_query()函数向 course 表插入数据。

答：主要代码如下：

```
$ sqlinsert = "insert into course values('c05192','oracle','必修',48,16,3)";
$ result = mysqli_query( $ conn1, $ sqlinsert) or die("插入失败".mysqli_error());
```

(3) 利用 mysqli_query()函数删除 course 表数据。

答：主要代码如下：

```
$ sqldelete = "delete from course where courseno = 'c05108'";
mysqli_query( $ conn1, $ sqldelete);
```

(4) 利用 mysqli_query()函数更新 course 表数据。

答：主要代码如下：

```
$ sql update = "update course  set cname = '机械零件' where  courseno = 'c06108'";
mysqli_query( $ conn1, $ sqlupdate);
```

(5) 查询教师号为 t05001 的教师信息，并利用输出命令 echo 和 print_r()函数两种方式输出。

答：PHP 代码和运行结果如下：

```
<?php
# 连接 MySQL 数据库服务器
$ conn1 = mysqli_connect("localhost","root","123456");
$ select = mysqli_select_db( $ conn1,"teaching");          //连接服务器中的 teaching
if( $ select)
   {
    header("Content - Type:text/html; charset = gb2312 ");     //设置字符集
     echo "数据库连接成功!" ;                                //判断是否连接成功
   }
$ sql = "select * from teacher where teacherno = 't05001';";
if ( $ result = mysqli_query( $ conn1, $ sql))
{
  while ( $ row = mysqli_fetch_assoc( $ result))
    {
      echo "<br />";
      echo "echo 格式: "." <br />";
      echo "教师号:     ". $ row['teacherno'];
      echo "      教师姓名: ". $ row['tname'];
      echo "       专业: ". $ row['major'];
      echo "       职称: ". $ row['prof'];
```

```
        echo  "       部门: ". $ row['department']." <br />";
        echo  "print_r()函数格式: "." <br />";
        print_r( $ row);
    }
    mysqli_free_result( $ result);                          //释放内存
    }
    mysqli_close( $ conn1);                                 //关闭连接对象
    ?>
```

运行结果如下:

数据库连接成功!
echo 格式:
教师号: t05001　教师姓名: 苏超然 专业: 软件工程　职称: 教授　　部门: 计算机学院
print_r()函数格式:
Array ([teacherno] => t05001 [tname] => 苏超然 [major] => 软件工程 [prof] => 教授 [department] => 计算机学院)

第15章 基于JSP技术的MySQL数据库应用开发实例

15.1 教学要求

15.1.1 基本要求

(1) 了解MySQL数据库在实例开发过程中的背景和意义。

(2) 了解系统开发的可行性分析过程。

(3) 掌握如何确定开发项目的目标的过程。

(4) 掌握如何利用软件工程的知识对系统进行需求分析的步骤。

(5) 了解数据字典开发的意义。

(6) 掌握如何设计数据库的概念结构和逻辑结构。

(7) 掌握E-R图和设计数据表的关系。

(8) 掌握利用JSP技术访问MySQL数据的基本操作。

(9) 掌握利用MySQL数据库设计在线考试系统的过程。

15.1.2 重点与难点

1. 本章重点

(1) MySQL数据库在实例开发过程中的背景和意义。

(2) 如何确定开发项目的目标的过程。

(3) 如何利用软件工程的知识对系统进行需求分析的步骤。

(4) 如何设计数据库的概念结构和逻辑结构。

(5) E-R图和设计数据表的关系。

(6) 利用JSP技术访问MySQL数据的基本操作。

2. 本章难点

(1) 如何确定开发项目的目标的过程。

(2) 如何利用软件工程的知识对系统进行需求分析的步骤。

(3) 如何设计数据库的概念结构和逻辑结构。

(4) E-R图和设计数据表的关系。

(5) 有关JSP技术的基本操作。

15.2 习题参考答案

1. 选择题

(1) 在关系数据库设计中,设计关系模式属于数据库设计的________。

A. 需求分析阶段　B. 概念设计阶段　C. 逻辑设计阶段　D. 物理设计阶段

(2) E-R 图提供了表示信息世界中实体、属性和________的方法。

A. 数据　B. 联系　C. 表　D. 模式

(3) E-R 图是数据库设计的工具之一,它一般适用于建立数据库的________。

A. 逻辑模型　B. 结构模型　C. 物理模型　D. 概念模型

(4) 将 E-R 图转换到关系模式时,实体与联系都可以表示成________。

A. 属性　B. 关系　C. 键　D. 域

(5) 如果关系模式 R 属于 1NF,且每个非主属性都完全函数依赖于 R 的主码,则 R 属于________。

A. 2NF　B. 3NF　C. BCNF　D. 4NF

本题答案:C B D B A

2. 思考题

(1) 简述 MySQL 数据库设计优化的基本过程。

答:优化数据库模型需要对 MySQL 适合的模型进行转换。转换的主要依据是 MySQL 数据库管理系统的功能及限制。

因此,数据库逻辑设计的结果不是唯一的。得到初步数据模型后,还应该适当地修改、调整数据模型的结构,以进一步提高 MySQL 数据库应用系统的性能,这就是数据模型的优化。关系数据模型的优化通常以规范化理论为指导,优化数据模型的方法主要体现在确定数据依赖。按需求分析阶段所得到的语义并不是规范化程度越高的关系就越优,当查询经常涉及两个或多个关系模式的属性时,系统必须经常地进行连接运算。对于一个具体应用来说,到底规范化进行到什么程度,需要权衡响应时间和潜在问题两者的利弊才能决定。

(2) 简述对于系统功能进行分析的基本要求。

答:任何一个软件系统的设计与开发都需要进行详细的需求分析,其目的就是尽量快速、准确、全面地获得系统的真实需求,规划出系统的整体功能,为系统的设计与实现做好完备而坚实的基础,使系统的开发工作得以顺利进行。

性能需求是指系统必须满足的定时约束或容量约束,通常体现在终端用户接入速率、响应时间、稳定性、可扩展性和并发用户支持等几个方面。具体说来包括如下三个方面。

① 可靠性。系统要求选用可靠的计算机及网络设备,数据库服务器等在条件允许的条件下还需采用磁盘镜像技术。数据库及操作系统软件要采用成熟的、能提供有效技术支持的主流产品,应用软件的设计编制应遵循规范化标准,整个项目的开发过程应得到有效监控。

② 安全性。系统的安全性是非常重要的,合理的安全控制可以使应用系统中的信息资

源得到有效的保护，故系统在数据库层和应用层都做了安全方面的设置：对所有系统用户需要有完善的口令加密功能，以保证系统及数据的安全性。应用软件对输入的数据进行合法性、有效性和完整性检验，如果输入数据存在问题，系统应能及时给予提示。

③ 易用性。系统的软硬件设计面向非专业的管理人员，管理系统具有美观友好的操作界面，使用简捷、易懂易学。

(3) 说明关系模型的基本术语的含义。

答：在关系数据库中，有以下几个常见的关系术语。

① 关系。关系就是一个二维表格，每个关系都有一个关系名，在 MySQL 中，一个关系称为一个表(Table)。

② 元组(记录)。在一个具体的关系中，表中的一行称为一个元组(记录)。

③ 属性(字段)。表中的一列即为一个属性(字段)，给每一个属性起一个名称即属性名。

④ 域。属性的取值范围，如成绩的范围一般为 0～100，性别的域是(男，女)等。

⑤ 码。也称为键。表中的某个属性组，它可以唯一确定一个元组，如学生学号，可以唯一确定一个具体的学生，也就成为学生关系的码。

⑥ 分量。元组中的一个具体的属性值，称为分量。

⑦ 关系模式。对关系的描述称为关系模式，一个关系模式对应一个关系，是命名的属性集合，一般表示为：

```
关系名(属性名 1,属性名 2,…,属性名 n)
```

第2部分
MySQL数据库的实验指导与实验安排

学习导读：

本部分包括本课程的实验指导和实验内容两方面内容。

实验指导部分包括 MySQL 的实验环境的进一步介绍，还有 MySQL 常用的两个可视化软件 Navicat for MySQL 和 phpMyAdmin 的基本操作，以及进行本课程实验的基本规范和建议。

实验内容共包括 16 个实验，实验 1～实验 14 的运行环境是 MySQL 5.7.17 软件和 MySQL Workbench 可视化软件，实验 15 的运行环境还需要 PHP 7.1 和 Apache 2.4 软件，实验 16 是综合性实验，利用 phpMyAdmin 软件，以"汽车销售管理数据库-arsales_info"为例，将 MySQL 的重点内容（如数据库、表及数据完整性、SQL 查询、视图、存储程序、导入导出、账户与权限管理、二进制文件管理等）操作了一遍，每一个实验的内容安排都与主教材的相关章节对应。

实验过程中，建议除了利用本书的代码外，还可以不断修改各种数据和参数进行实验操作。在实验过程中，对于可视化软件的操作，可以多尝试操作一下教材中因篇幅关系未介绍的操作。另外，读者不需要熟悉每一种可视化软件，熟悉其中的一种即可。

第 16 章　MySQL 数据库的实验操作环境

MySQL 数据库的实验操作相对于 Oracle 和 SQL Server，需要更加熟悉实验环境。Oracle 功能强大，数据处理能力强，有着功能完善的企业管理器，对于初学者有很好的帮助。SQL Server 采用集成化界面设计，将查询分析器、企业管理器和对象资源管理器等工具集成在一起，尤其是其可视化操作性能强大，几乎能够实现所有基本操作。

MySQL 数据库的基本操作相对简单一些，实验中常用到的工具主要包括安装完成后自带的命令工具。另外，熟悉 MySQL 实验环境的配置和数据库操作过程中常见的错误代码也是实验过程中不可分割的环节。

本章将从实验操作的角度，详细介绍 MySQL 数据库中的自带命令工具的操作、MySQL 命令的执行、常用的 MySQL 数据库的手工配置和实验过程中的常见错误代码标识等内容。

16.1　MySQL 的自带命令

安装 MySQL 数据库成功后，会在安装目录中找到 bin 文件夹，该文件夹包含了可执行文件，如 mysql. exe、mysqld. exe 和 mysqladmin. exe 等。下面将介绍一些常见的命令，并且使用它们进行简单的操作。

16.1.1　查看自带的命令工具

查看 MySQL 数据库自带的命令工具很简单，找到 MySQL 数据库的文件安装路径下的 bin 文件夹并打开，如图 16-1 所示。

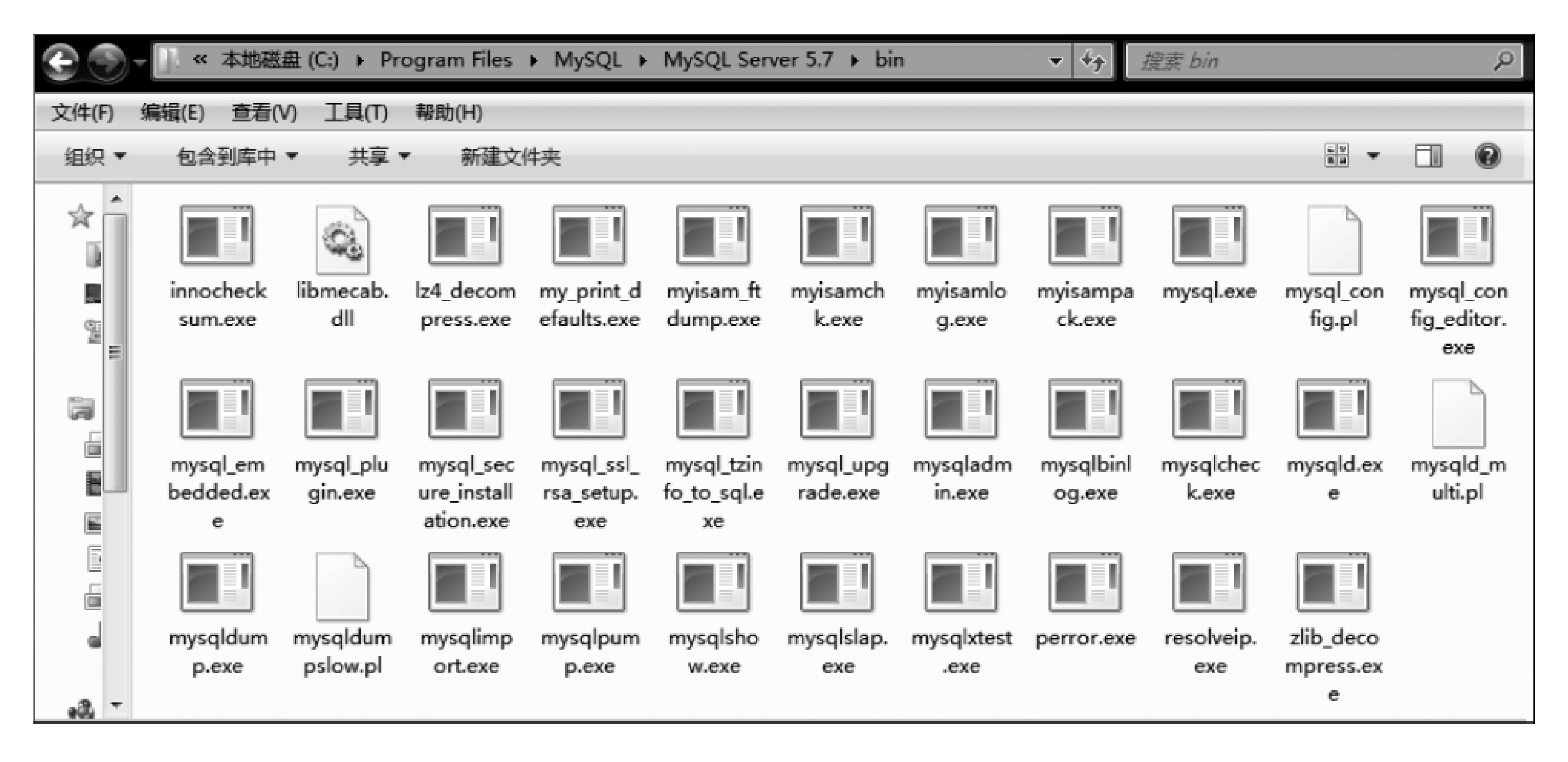


图 16-1　bin 文件夹下的命令

从图 16-1 中可以看出，MySQL 数据库中自带了多个命令工具，通过执行这些命令可以实现不同的操作，大多数的操作都是通过 mysql 命令来实现的。另外，mysqladmin 命令主要用来显示服务器状态，mysqlimport 使用 load data infile 将文本文件导入相关表的客户程序。表 16-1 对图 16-1 中的常用工具进行了简单说明。

表 16-1　MySQL 中的常用命令工具

命令工具	说　　明
myisampack	压缩 MyISAM 表以产生更小的只读表的一个工具
mysql	交互式输入 SQL 语句或从文件以批处理模式执行它们的命令行工具
mysqladmin	执行管理操作的客户程序，例如管理数据库、重载授权表、将表保存到硬盘上，以及重新打开日志文件。mysqladmin 还可以用来检索版本、进程，以及服务器的状态信息
mysqlbinlog	从二进制日志读取语句的工具。在二进制日志文件中包含执行过的语句的日志可用来帮助从崩溃中恢复
mysqlcheck	检查、修复、分析以及优化表的表维护客户程序
mysqldump	将 MySQL 数据库转储到一个文件(例如 SQL 语句或 Tab 分隔符文本文件)的客户程序
mysqlimport	使用 load data infile 将文本文件导入相关表的客户程序
mysqlshow	显示数据库、表、列以及索引相关信息的客户程序
perror	显示系统或 MySQL 错误代码含义的工具

16.1.2　DOS 窗口执行命令

表 16-1 所列出的命令通常都是在 DOS 窗口中执行操作的，运行这些命令时有以下两种方式：直接进入安装路径的目录的 bin 目录下，然后运行命令查看；将路径添加到 Path 环境变量中，然后直接在 DOS 窗口中执行命令。

1. 直接进入安装路径的 bin 目录

这些命令实现的过程是：执行“开始”→“运行”命令，弹出“运行”对话框，输入 cmd 后按 Enter 键打开 DOS 窗口，效果如下：

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。
C:\Users\Administrator>
```

打开 DOS 窗口后首先执行 CD 命令进入 MySQL 数据库 bin 文件夹，结果如下：

```
C:\Users\Administrator>CD C:\Program Files\MySQL\MySQL Server 5.7\bin
C:\Program Files\MySQL\MySQL Server 5.7\bin>
```

进入 MySQL 安装路径下的 bin 目录后，直接执行 mysql 命令，并以 root 的身份登录 MySQL 管理系统。输出结果如下：

```
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -u root -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.17-log MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

如果用户想要退出当前命令并执行其他命令(如 mysqladmin),执行 exit 命令退出当前命令,重新回到 bin 文件夹。输出结果如下:

```
mysql> quit
Bye
C:\Program Files\MySQL\MySQL Server 5.7\bin>
```

2. 将路径添加到 Path 环境变量中

通过第 1 种方式虽然可以执行命令语句,但是它相当烦琐,如果开发者关闭当前 DOS 窗口打开一个新窗口时,还需要重新进入安装目录。可以将安装路径添加到 Path 环境变量中,然后在 DOS 窗口中直接执行命令运行。具体实现的过程可以参看教材第 1 章的内容。

系统变量设置完成后,重新启动计算机后打开 DOS 窗口。直接执行 mysqlshow 命令进行操作,输出结果如下:

```
C:\Users\Administrator> mysqlshow -u root -p
Enter password: ******
   +-----------------+
   |    Databases    |
   +-----------------+
   | information_schema |
   | booksmgt        |
   | db_examsystem   |
   | mysql           |
   | mysqltest       |
   | performance_schema |
   | sakila          |
   | student         |
   | sys             |
   | teaching        |
   | world           |
   +-----------------+
```

16.1.3 mysqladmin 命令

mysqladmin 命令与 mysql 命令一样,经常会使用到,它主要用来对数据库做一些简单操作,以及显示服务器状态等。mysqladmin 命令的基本语法格式如下:

```
mysqladmin [option] command [command option] command ...
```

直接在 DOS 窗口中执行 mysqladmin 语句或者输入"mysqladmin -?"可以打印出它的帮助信息。

1. mysqladmin 命令的相关选项

帮助信息包括 3 部分,第 1 部分为 mysqladmin 命令的常用选项,表 16-2 对常用的选项

进行了说明。

表 16-2 mysqladmin 命令的常用选项

选 项 名 称	说 明
-c，--count=#	自动运行次数，必须和-i 一起使用
-f，--force	禁用错误，drop 数据库时不提示，执行多条命令时出错继续执行
-C，--compress	在服务器和客户端之间使用压缩
--character-sets-dir=name	设置字符集目录
--default-character-set=name	设置字符集默认目录
-?，--help	显示帮助信息
-h，--host=name	主机名
-p，--password[=name]	服务器连接密码
-P，--port=#	端口号
--protocol=name	设置连接使用的协议
-s，--silent	如果连接服务器失败则退出
-u，--user=name	用户名
-V，--version	显示 mysqladmin 的版本
-E，--vertical	垂直显示输出
-w，--wait[=#]	如果连接断开，则等待并重试
--connect_timeout=#	连接超时
--shutdown_timeout=#	超时关闭

【例 16-1】 直接打开 DOS 窗口，并且输入"mysqladmin -V"查看当前 mysqladmin 的版本。

代码和运行结果如下：

```
C:\Users\Administrator>mysqladmin -V
mysqladmin  Ver 8.42 Distrib 5.7.17, for Win32 on AMD64
```

2. mysqladmin 的可执行命令

mysqladmin 帮助信息的第 2 部分表示 mysqladmin 的相关变量。第 3 部分表示 mysqladmin 可以执行的相关服务器命令，目前 mysqladmin 支持的命令有创建新的数据库、删除一个数据库，以及清除所有缓存的主机和所有日志等，表 16-3 针对这些命令进行了详细说明。

表 16-3 mysqladmin 可以执行的命令

命 令 名 称	说 明
create databasename	创建一个新数据库
debug	用于通知服务器将调试信息写入日志
drop databasename	删除一个数据库及其所有表
extended-status	显示服务器的一个扩展状态信息
flush-hosts	清除所有缓存的主机
flush-logs	清除所有日志

续表

命令名称	说明
flush-status	清除状态变量
flush-tables	清除所有表
flush-threads	清除线程缓存
flush-privileges	再次装载授权表(同 reload)
kill id,id,...	杀死 mysql 线程
password[new-password]	将旧口令改为新口令
old-password[new-password]	修改口令
ping	检查 mysqld 是否在线
processlist	显示服务器中的活跃线程列表
reload	重载授权表
refresh	刷新所有表并关闭和打开日志文件
shutdown	关掉服务器
status	显示服务器的简短状态消息
start-slave	启动 slave
stop-slave	关闭 slave
variables	打印出可用变量
version	得到服务器的版本信息

【例 16-2】 直接在 mysqladmin 命令后跟 ping 命令检查 mysqld 是否在线。

代码和运行结果如下：

```
C:\Users\Administrator>mysqladmin -u root -p ping
Enter password: ****
mysqld is alive
```

【例 16-3】 在 DOS 窗口中执行 mysqladmin 命令时，也可以通过执行 version 命令来查看详细信息。

代码和运行结果如下：

```
C:\Users\Administrator>mysqladmin -u root -p version
Enter password: ******
mysqladmin  Ver 8.42 Distrib 5.7.17, for Win32 on AMD64
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.
Server version          5.7.17-log
Protocol version        10
Connection              localhost via TCP/IP
TCP port                3306
Uptime:                 8 hours 21 min 13 sec
Threads: 1  Questions: 14  Slow queries: 0  Opens: 108  Flush tables: 1  Open tables: 101
Queries per second avg: 0.000
C:\Users\Administrator>
```

【例 16-4】 通过执行 mysqladmin 命令语句查看 MySQL 的状态值。

代码和运行结果如下：

```
C:\Users\Administrator>mysqladmin proc stat -u root -p
Enter password: ******
    +---+----+--------+---+------+-----+--------+--------+
    | Id  | User| Host         | db | Command| Time | State      | Info          |
    +---+----+--------+---+------+-----+--------+--------+
    | 9  |root |localhost:62022 | | Query |0| starting |show processlist |
    +---+----+--------+---+------+-----+--------+--------+
Uptime: 30623  Threads: 1  Questions: 17  Slow queries: 0  Opens: 108  Flush tables: 1  Open
tables: 101  Queries per second avg: 0.000
```

执行命令完毕后都会输出状态信息。下面具体对输出的各项信息进行说明：

- Uptime：表示 MySQL 服务器已经运行的秒数。
- Threads：活动线程(客户)的数量。
- Questions：从 mysqld 启动以来客户问题(查询)的数量。
- Slow Query：执行时间超过 long_query_time 秒的查询数量。
- Opens：服务器已经打开数据库表的数量。
- Flush tables：服务器已经执行的 flush、refresh 和 reload 命令的数量。
- Open tables：目前打开的表的数量。
- Queries per second avg：执行平均用时秒数。

16.1.4 mysqlshow 命令

使用 mysqlshow 命令可以很快地查找存在哪些数据库、数据库中的表，以及表中的列或者索引。基本语法如下：

```
mysqlshow[options] [db_name [tal_name [col_name]]]
```

在上述所示的语法中，如果没有指定数据库，则显示所有匹配的数据库；如果没有指定数据库表，则显示数据库中所有匹配的表；如果没有给出列，则显示表中所有匹配的列和列类型。

直接执行 mysqlshow 或“mysqlshow -?”命令可以查看该命令的帮助信息，输出的信息包括两部分：第 1 部分显示 mysqlshow 命令的相关选项，常用选项的说明如表 16-4 所示。

表 16-4 mysqlshow 命令的相关选项

选 项 名 称	说 明
---help,-?	显示一个帮助消息并退出
-C，--compress	压缩所有的客户和服务器之间发送的信息(如果它们都支持压缩)
-h，--host=name	在显示的主机上连接 MySQL 服务器
-k，--keys	显示表索引
-i，--status	显示关于每个表的额外信息
-p，--password[=name]	连接服务器时使用的密码。如果使用短选项形式(-p)，不能在选项和密码之间存在空格。如果在命令行中忽略了--password 或-p 选项后面的密码值，将提示输入一个密码

续表

选项名称	说　　明
-u, --user=name	连接服务器时使用的 MySQL 用户名
-v, --verbose	冗长模式,打印出程序操作的详细信息。此选项可以多次使用以便增加信息总量
-V, --version	显示版本信息并退出

例如,直接在 DOS 窗口中执行"mysqlshow mysql -V"命令,输出结果如下:

```
C:\Documents and Settings\Administrator> mysqlshow mysql  - V
mysqlshow  Ver 9.10 Distrib 5.6.13, for Win32 (x86)
```

【例 16-5】 演示 mysqlshow 命令的简单使用。

(1) 在 DOS 窗口中执行 mysqlshow 命令,执行完毕后会显示所有的数据库和表。前面利用该语句查看过数据库,也可以查看指定数据库中的所有表和视图。

代码和运行结果如下:

```
C:\Users\Administrator> mysqlshow teaching  - u root  - p
Enter password: ******
   Database: teaching
    +-----------+
    |   Tables   |
    +-----------+
    | bef_after     |
    | course        |
    | course_avg    |
    | de_teacher    |
    | mytest        |
    | score         |
    | se_course     |
    | stu_score     |
    | stu_view2     |
    | student       |
    | teach_course  |
    | teach_view1   |
    | teacher       |
    | v_avgstu      |
    | v_dept        |
    | v_sex         |
    | v_teacher     |
    +-----------+
```

(2) 在数据库之后指定选项-v,会输出程序操作的详细信息。更改前面的命令,为上述命令指定-v 选项。

代码和运行结果如下:

```
C:\Users\Administrator> mysqlshow teaching  - v  - u root  - p
Enter password: ******
   Database: teaching
```

```
+----------+-------+
|  Tables  | Columns |
+----------+-------+
| bef_after    |      3  |
| course       |      6  |
| course_avg   |      2  |
| de_teacher   |      2  |
| mytest       |      1  |
| score        |      4  |
| se_course    |      6  |
| stu_score    |      5  |
| stu_view2    |      5  |
| student      |      7  |
| teach_course |      2  |
| teach_view1  |      5  |
| teacher      |      5  |
| v_avgstu     |      3  |
| v_dept       |      5  |
| v_sex        |      5  |
| v_teacher    |      5  |
+----------+-------+
17rows in set.
```

(3) 利用 mysqlshow 命令的相关语句查看数据库表 se_course,并使用选项-v 指定打印出程序操作的详细信息。

代码和运行结果如下:

```
C:\Users\Administrator>mysqlshow teaching se_course -v -u root -p
Enter password: ******
Database: teaching  Wildcard: se_course
+--------+--------+
|  Tables  | Columns  |
+--------+--------+
| se_course |      6  |
+--------+--------+
1 row in set.
```

16.1.5 mysqlbinlog 命令

mysqlbinlog 命令是用于处理二进制日志文件的实用工具,服务器生成的二进制日志文件写成二进制格式,如果要检查这些文本格式的文件,可以使用 mysqlbinlog 命令工具。

mysqlbinlog 命令语法如下:

```
mysqlbinlog [options] log-files...
```

开发者可以直接在 DOS 窗口中执行 mysqlbinlog 或"mysqlbinlog -?"命令查看帮助信息。该命令显示的帮助信息包括两部分:第 1 部分显示 mysqlbinlog 命令的相关选项,常用选项如表 16-5 所示;第 2 部分显示了 mysqlbinlog 命令的相关变量。

表 16-5　mysqladmin 执行的相关选项

选项名称	说　　明
-?，--help	显示帮助信息并退出
-d，--database＝name	只列出该数据库的条目(只用于本地)
-f，--force -read	使用该选项，如果 mysqlbinlog 读它不能识别的二进制日志事件，它会打印警告，忽略该事件并继续。没有该选项，如果 mysqlbinlog 读到此类事件则停止
-H，--hexdump	在注释中显示日志的十六进制转储
-h，--host＝name	获取给定主机上的 MySQL 服务器的二进制日志
-l，--local-load＝name	为指定目录中的 load data infile 预处理本地临时文件
-P，--port＝#	用于连接远程服务器的 TCP/IP 端口号
--protocol＝name	使用的连接协议
-R，--read-from-remote -server	从 MySQL 服务器读二进制日志。如果未给出该选项，任何连接参数选项将被忽略。这些选项是- -host、- -password、- -port、- -protocol、--socket 和--user
-S，--socket＝name	用于连接的套接字文件
--start-datetime＝name	从二进制日志中第 1 个日期时间等于或晚于 datetime 参量的事件开始读取。datetime 值相对于运行 mysqlbinlog 的机器上的本地时区。该值格式应符合 DATETIME 或 TIMESTAMP 数据类型
--stop-datetime＝name	从二进制日志中第 1 个日期时间等于或晚于 datetime 参量的事件起停止读。关于 datetime 值的描述参见--start-datetime 选项

通常情况下，可以使用 mysqlbinlog 命令直接读取二进制日志文件，并将文件用于本地 MySQL 服务器，也可以使用--read-from-remote-server 选项从远程服务器读取二进制日志文件。当读取远程二进制日志时，可以通过连接参数选项来指示如何连接服务器，但是它们经常被忽略，除非还指定了--read-from-remote-server 选项。还可以使用 mysqlbinlog 来读取在复制过程中从服务器所写的中继日志文件，中继日志格式与二进制日志文件相同。

【例 16-6】 利用 mysqlbinlog 命令输出 PGIG1MIWMYPOFBS-bin. 000005 文件中包含的所有语句以及其他信息。如果指定的文件不存在，则会输入提示信息，提示该文件并不存在。

代码和运行结果如下：

```
C:\Users\Administrator> mysqlbinlog PGIG1MIWMYPOFBS - bin. 000005
/ * !50530 SET @@SESSION. PSEUDO_SLAVE_MODE = 1 * /;
/ * !50003 SET @OLD_COMPLETION_TYPE = @@COMPLETION_TYPE, COMPLETION_TYPE = 0 * /;
DELIMITER / * ! * /;
mysqlbinlog: File 'PGIG1MIWMYPOFBS - bin. 000045 ' not found (Errcode: 2 - No such file or
directory)
SET @@SESSION. GTID_NEXT = 'AUTOMATIC' / * added by mysqlbinlog * / / * ! * /;
DELIMITER ;
# End of log file
/ * !50003 SET COMPLETION_TYPE = @OLD_COMPLETION_TYPE * /;
/ * !50530 SET @@SESSION. PSEUDO_SLAVE_MODE = 0 * /;
```

16.1.6 perror 命令

perror 命令用于解释错误命令，对于大多数系统错误，除了内部文本信息之外，MySQL 还会按下面的风格显示系统错误代码。

```
message ... (Errcode:# )
message ... (errorno:# )
```

通过检查系统文档或使用 perror 工具，可以检查错误代码的意义。perror 为系统错误代码或存储引擎（表处理）错误代码打印其描述信息。

perror 命令的基本语法如下：

```
perror [options] errorcode
```

【例 16-7】 执行 perror 命令显示的错误信息。

代码和运行结果如下：

```
C:\Users\Administrator>perror 9 19
OS error code    9:   Bad file descriptor
Win32 error code 9: 存储控制块地址无效。
OS error code   19:   No such device
Win32 error code 19: 介质受写入保护。
```

执行 perror 命令查看帮助信息，输出的帮助信息如下：

```
C:\Users\Administrator>perror
perror Ver 2.11, for Win32 (AMD64)
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respectiveowners.
Print a description for a system error code or a MySQL error code.
If you want to get the error for a negative error code, you should use
-- before the first error code to tell perror that there was no more options.
Usage: perror  [OPTIONS] [ERRORCODE [ERRORCODE...]]
   -?, --help        Displays this help and exits.
   -I, --info        Synonym for --help.
   -s, --silent      Only print the error message.
   -v, --verbose    Print error code and message (default).
                   (Defaults to on; use --skip-verbose to disable.)
   -V, --version    Displays version information and exits.
Variables (--variable-name=value)
and boolean options {FALSE|TRUE}   Value (after reading options)
---------------------------------- --------------------------------------
verbose                          TRUE
```

由此帮助信息可以看出，perror 命令与前面的其他命令相比，比较简单。

16.2 MySQL 的常用命令

MySQL 是一个简单的 SQL 工具，支持交互式和非交互式使用。交互使用时查询结果采用 ASCII 表格式。当采用非交互式模式时，例如用作过滤器，其结果为 Tab 分隔符格式。

16.2.1 mysql 命令的语法格式

mysql 命令的基本语法如下：

```
mysql [options] [database]
```

在 DOS 窗口中可以直接执行 mysql 或“mysql -?”命令语句查看帮助信息，表 16-6 对帮助信息中常用的一些选项进行了介绍。

表 16-6 mysql 命令执行的相关选项

选项名称	说明
-?, --help	显示帮助信息并退出
-B, --batch	打印结果，使用 Tab 作为列间隔符，每个行占用新的一行。使用该选项，则 mysql 不使用历史文件
--character-sets-dir=name	字符集的安装目录
-D, --database=name	要使用的数据库，主要在选项文件中使用
-T, --debug-info	当程序退出时输出部分调试信息
--default-character-set=name	使用 name 作为默认字符集
-E, --vertical	垂直输出查询输出的行。如果没有该选项，可以使用\G 结尾指定单个语句的垂直输出
-f, --force	即使出现一个 SQL 错误仍继续
-h, --host=name	连接指定主机上的 MySQL 服务器
-H, --html	产生 HTML 输出
-N, --skip-column-names	在结果中不写列名
-s, --silent	沉默模式，产生少的输出。可以多次使用该选项以产生更少的输出
-X, --xml	产生 XML 输出
-n, --unbuffered	每次查询后刷新缓存区
-u, --user=name	连接服务器时 MySQL 使用的用户名
-v, --verbose	冗长模式。产生更多的输出，可以多次使用该选项以产生更多的输出
-V, --version	显示版本信息并退出
-w, --wait	如果不能建立连接，等待并重试而不是放弃
--show-warnings	如果每个语句后有警告则显示，此选项适用于交互式和批处理模式
--net-buffer-length=#	TCP/IP 和套接字通信缓冲区大小
--max-allowed-packet=#	从服务器发送或接收的最大包长度
--connect-timeout=#	连接超时前的秒数

例如，直接在 DOS 窗口中执行“mysql -V”命令查看版本信息，输出结果如下：

```
C:\Users\Administrator>mysql  -V
mysql  Ver 14.14 Distrib 5.7.17, for Win32 (MAD64)
```

16.2.2 MySQL 的命令

MySQL 的命令将发出的 SQL 语句发送到待执行的服务器，还有一系列命令 MySQL 可以自己解释。如果要查看这些命令，在 mysql>提示符下输入 help 或\h 语句，输出结果如下：

```
mysql > \h
For information about MySQL products and services, visit:
   http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
   http://dev.mysql.com/
To buy MySQL Enterprise support, training, or other products, visit:
   https://shop.mysql.com/
List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?         (\?) Synonym for `help'.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
ego       (\G) Send command to mysql server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to mysql server.
help      (\h) Display this help.
notee     (\t) Don't write into outfile.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source    (\.) Execute an SQL script file. Takes a file name as an argument.
status    (\s) Get status information from the server.
tee       (\T) Set outfile [to_outfile]. Append everything into given outfile.
use       (\u) Use another database. Takes database name as argument.
charset   (\C) Switch to another charset. Might be needed for processing binlogwith multi-byte
charsets.
warnings  (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.
resetconnection(\x) Clean session context.
For server side help, type 'help contents'
```

MySQL 的每个命令都有长形式和短形式，长形式对大小写不敏感；而短形式则需要区分。长形式后面可以加一个分号结束符，但是短形式不可以。

【例 16-8】 status 命令提供连接和使用的服务器相关的部分信息。

代码和运行结果如下：

```
mysql > status
------------
mysql   Ver 14.14 Distrib 5.7.17, for Win32 (AMD64)
Connection id:          23
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.7.17-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    utf8
```

```
Db      characterset:    utf8
Client characterset:     gbk
Conn.   characterset:    gbk
TCP port:               3306
Uptime:                 11 hours 39 min 24 sec
Threads: 1  Questions: 90  Slow queries: 0  Opens: 125  Flush tables: 1  Open tables: 118
Queries per second avg: 0.002
```

若执行 quit 或 exit 语句则退出 MySQL 执行其他命令的状态，返回 DOS 界面。

有些情况下，当被查询数据库表中的数据过多时，水平显示这些数据会显得杂乱无章，这时使用垂直显示更加容易读取。使用\G 而不用分号结束查询可以垂直显示查询。

【例 16-9】 查询 teaching 数据库下 se_course 表中所有列的值，并将结果以垂直方式输出。

代码和运行结果如下：

```
mysql > SELECT * FROM se_course \G
      ****************** 1. row ******************
         SC_no: 1
     studentno: 19120000111
      courseno: co1236
     teacherno: t01237
         score: NULL
       SC_time: 2017 - 05 - 01 18:40:23
      ****************** 2. row ******************
         SC_no: 2
     studentno: 19120000222
      courseno: co1237
     teacherno: t01239
         score: NULL
       SC_time: 2017 - 05 - 05 18:24:22
     2 rows in set (0.05 sec)
```

16.2.3 从文本文件执行 MySQL 命令

通过 DOS 窗口和 MySQL 控制台都可以执行 MySQL 语句进行一般交互。例如，将 MySQL 语句放到一个文件中，然后从该文件中进行读取。要实现这种方式，可以创建一个文本文件，并且该文件中包含要执行的语句，然后进行调用。

shell 命令的调用格式如下：

```
shell > mysql database_name < text_file
```

还可以用一个 use database_name 语句启动文本文件。此时，不需要在命令行中指定数据库名，语法如下：

```
shell > mysql < text_file
```

如果正在运行 MySQL，可以使用 source 或\. 语句执行 SQL 脚本文件，语法如下：

```
mysql > source filename
mysql > \.Filename
```

【例 16-10】 在 D 磁盘下创建名称是 mytest 的文本文件。文本内容如下：

```
use teaching;
select  * from se_course;
select now();
```

在 DOS 窗口中执行 MySQL 命令语句，指定文本文件并执行该文件中的语句。

输出结果如下：

```
C:\Users\Administrator>mysql<D:\mytest.txt  -u root  -p
Enter password: ******
SC_no   studentno      courseno       teacherno      score   SC_time
1       19120000111    co1236   t01237   NULL     2017-05-01 18:40:23
2       19120000222    co1237   t01239   NULL     2017-05-05 18:24:22
now()
2017-06-29 20:07:48
C:\Users\Administrator>
```

16.3 MySQL 实验中的常见操作

MySQL 数据库安装完成后，可能会根据实际情况更改 MySQL 数据库的某些配置。一般可以通过手动方式更改 MySQL 数据库的某些配置。手动更改方式虽然比较困难，但是这种配置方式更加灵活。

安装 MySQL 数据库时提到将 MySQL 数据库安装到 C 磁盘下，文件安装在 C:\Program Files\MySQL\MySQL Server 5.7 目录下，如图 16-2 所示。数据库文件安装在 C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server 5.7 目录下，该目录下包含 data 文件夹和 my.ini 文件。

从图 16-2 中可以看出，安装文件包含多个文件夹和 my-default.ini 文件。其中，bin 文件夹下都是可执行文件，例如 mysql.exe、mysqlId.exe 和 mysqladmin.exe 等；docs 文件夹自带了两个关于 mysql5.7 的英文说明文档。include 文件夹下都是头文件，如 decimal.h、errmsg.h 和 mysql.h 等；lib 文件夹下都是库文件，该文件夹下有 plugin 和 debug 两个文件夹；share 文件夹下是字符集、语言等信息。



图 16-2 MySQL 的安装文件

my-default.ini 文件里面存放的是一些默认配置信息，而数据库安装目录下的 my.ini 文件才是真正的配置文件。因此，只要修改 my.ini 文件中的内容就可以达到更改配置的目的。my.ini 文件中包含了大量的内容，下面介绍该文件时会去掉大量的注释内容，然后从客户端配置和服务器端配置进行介绍。

如果安装时选择的配置不一样，那么配置文件就会稍有不同。通常情况下，经常修改的是默认字符集、默认存储引擎和端口等信息，其他参数修改比较复杂，一般不进行修改。另外，每次修改参数后，必须重新启动 MySQL 服务才会有效。

1. 客户端配置

客户端的配置非常简单，port 参数表示 MySQL 数据库的端口，默认端口是 3306。default-character-set 参数是客户端的默认字符集，现在设置的参数是 utf8。如果想要更改客户端的设置内容，可以直接在 my.ini 文件中进行更改，部分配置内容如下：

```
# Other default tuning values
# MySQL Server Instance Configuration File
# ----------------------------------------------------------------------
# Generated by the MySQL Server Instance Configuration Wizard
# 该文件是使用 MySQL 配置向导生成的
# CLIENT SECTION
# ----------------------------------------------------------------------
# 下面将会是客户端的各个参数的介绍。[client]和[mysql]都是客户端的

[client]
no-beep

# pipe
# socket = mysql
port = 3306

[mysql]
default-character-set = utf8
```

2. 服务器端配置

与客户端相比，服务器端的配置要复杂得多，自动生成的文件内容是通过英文解释说明的，下面通过中文对相关的内容进行了解释说明。

```
# SERVER SECTION
# ----------------------------------------------------------------------
# 下面是服务器端各个参数的介绍。[mysqlId]表示下面的内容属于服务器端
# server_type = 3
[mysqld]
# port 参数表示 MySQL 数据库的端口,默认端口是 3306
port = 3306
# basedir 参数表示 MySQL 安装路径,此处显示的安装路径为 F 磁盘
# basedir = "C:/Program Files/MySQL/MySQL Server 5.7/"
# datadir 参数表示 MySQL 数据文件的存储位置
datadir = "C:/Documents and Settings/All Users/Application Data/MySQL/MySQL Server 5.7/data\"
# character-set-server 参数表示默认的字符集,这个字符集是服务器端的
character-set-server = utf8
```

```
# default - storage - engine 参数表示默认的存储引擎,存储引擎表示数据的存储方式
default - storage - engine = INNODB
# sql - mode 参数表示 SQL 模式的参数,通过这个参数,可以设置检验 SQL 语句的严格程度
sql - mode = "STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"
# plugin - load 参数表示 Windows 身份验证
# plugin - load = authentication_windows.dll
# 通用查询日志和慢查询日志
log - output = NONE
general - log = 0
general_log_file = "WS.log"
slow - query - log = 0
slow_query_log_file = "WS - slow.log"
long_query_time = 10
# 二进制日志
# log - bin
# 错误日志
log - error = "WS.err"
# max_connections 参数表示允许同时访问 MySQL 服务器的最大连接数。其中一个连接将保留作为
#管理员登录
max_connections = 100
# query_cache_size 参数表示查询时的缓存大小。缓存中可以存储以前的 SELECT 语句查询过的信
#息,遇到相同的查询时,可以直接从缓存中取出结果
query_cache_size = 1M
# table_open_cache 参数表示所有进程打开表的总数
table_open_cache = 2000
# tmp_table_size 参数表示内存中临时表的最大值
tmp_table_size = 5M
# thread_cache_size 参数表示保留客户端线程的缓存
thread_cache_size = 9
```

3. InnoDB 存储引擎使用的参数

InnoDB 是 MySQL 5.7 数据库的默认存储引擎，my.ini 文件中还包含 InnoDB 和 MyISAM 存储引擎使用的参数。下面的内容是 InnoDB 存储引擎的一些常用参数。

```
# *** INNODB Specific options ***
# innodb_data_home_dir = 0.0
# innodb_additional_mem_pool_size
                              //参数表示附加的内存池,用来存储 InnoDB 表的内容
innodb_additional_mem_pool_size = 2M
# innodb_flush_log_at_trx_commit      //参数设置提交日志的时机
innodb_flush_log_at_trx_commit = 1
# innodb_log_buffer_size              //参数表示用来存储日志数据的缓冲区的大小
innodb_log_buffer_size = 1M
# innodb_buffer_pool_size
                     //表示缓存的大小。InnoDB 使用一个缓冲池来保存索引和原始数据
innodb_buffer_pool_size = 8M
# innodb_log_file_size                //参数表示日志文件的大小
innodb_log_file_size = 48M
# innodb_thread_concurrency           //参数表示在 InnoDB 存储引擎允许的线程最大数
innodb_thread_concurrency = 8
```

这里只列出了 InnoDB 存储引擎的一些常用参数。当然，my.ini 文件中的内容并不是前面所列出的那些。my.ini 文件中的内容是英文的，上面加上了中文说明，可以找到安装目录下的 my.ini 文件并打开查看其所有内容。

16.4 常见的错误代码和处理

在客户端的控制台执行语句时，如果执行的语句不合法或者错误，则会输出有关的错误信息。例如，在命令中执行“select month(current_data);”语句时出现 1054 错误。

代码和输出结果如下：

```
mysql> select  month(current_data);
ERROR 1054 (42S22): Unknown column 'current_data ' in 'field list'
```

上述提示 1054，表示 current_data 列不存在。除了这些错误外，在执行语句时可能还会出现其他的错误，表 16-7 列出了一些常见的错误代码，并且对这些代码进行了说明。在实验过程中，可以根据错误代码查看错误原因。

表 16-7 执行 MySQL 语句时的常见错误代码

错误代码	说明
1005	创建表失败
1006	创建数据库失败
1007	数据库已存在，创建数据库失败
1008	数据库不存在，删除数据库失败
1009	不能删除数据库文件导致删除数据库失败
1010	不能删除数据目录导致删除数据库失败
1011	删除数据库文件失败
1012	不能读取系统表中的记录
1045	不能连接数据库，用户名或密码错误
1048	字段不能为空
1049	数据库不存在
1050	数据表已存在
1051	数据表不存在
1054	字段不存在
1065	无效的 SQL 语句，SQL 语句为空
1081	不能建立 Socket 连接
1114	数据表已满，不能容纳任何记录
1116	打开的数据表太多
1129	数据库出现异常，请重启数据库
1130	连接数据库失败，没有连接数据库的权限
1133	数据库用户不存在
1141	当前用户无权访问数据库
1142	当前用户无权访问数据表
1143	当前用户无权访问数据表中的字段
1146	数据表不存在

续表

错误代码	说　　明
1147	未定义用户对数据表的访问权限
1149	SQL 语句语法错误
1158	网络错误，出现读错误，请检查网络连接状况
1159	网络错误，读超时，请检查网络连接状况
1160	网络错误，出现写错误，请检查网络连接状况
1161	网络错误，写超时，请检查网络连接状况
1162	字段值重复，入库失败
1169	字段值重复，更新记录失败
1177	打开数据表失败
1180	提交事务失败
1181	回滚事务失败
1203	当前用户和数据库建立的连接已到达数据库的最大连接数，请增大可用的数据库连接数或重启数据库
1216	外键约束检查失败，更新子表记录失败
1217	外键约束检查失败，删除或修改主表记录失败

第 17 章　MySQL 数据库的常用可视化软件

MySQL 数据库的基本操作相对简单一些,实验中常用到的工具主要包括安装完成后自带的命令工具,这些命令多数需要记忆,操作时显得比较复杂和麻烦。

在 MySQL 中还可以使用图形界面管理工具方便地进行数据库操作。MySQL 的图形管理工具有很多,常用的有 MySQL GUI Tools、MySQL Workbench、phpMyAdmin 和 Navicat 等。教材中主要使用 MySQL Workbench 工具进行 MySQL 数据库的可视化操作。MySQL Workbench 是一款专为 MySQL 设计的 E-R/数据库建模工具。

本章将介绍 Navicat 和 phpMyAdmin 可视化工具的主要应用。

17.1　可视化工具 Navicat 软件的使用

Navicat 软件是一套快速、可靠且价格适宜的 MySQL 数据库管理和开发工具。Navicat 易学易用,支持中文,特别适合初学者进行数据库管理操作,也有免费版本提供。

Navicat 适用于 Windows 和 Linux 等多种操作系统,可以让用户连接到任何本机或远程服务器,提供一些实用的数据库工具如数据操作、数据传输、导入导出、备份还原、报表创建工具及计划以协助管理数据等。

Navicat 工具包含多个成员,主要包括 Navicat Premium、Navicat for MySQL、Navicat for SQL Server 和 Navicat for Oracle 等。可以从 Navicat 的官方网站 https://www.navicat.com /en/download /navicat-for-mysql 下载不同操作系统下的 Navicat 版本。

Navicat 是一个可多重连接的数据库管理工具,它可连接到 MySQL、Oracle、PostgreSQL、SQLite、SQL Server 等数据库,使得管理不同类型的数据库更加方便。

Navicat 符合专业开发人员的所有需求,对维护数据库服务器的新手来说又相当容易学习,它具有极完备的图形用户界面,可让用户简便地以安全且简单的方法创建、组织、访问和共享信息。

17.1.1　Navicat 安装和连接

1. Navicat 的安装

首先从指定网站下载 Navicat 软件,例如 mysqlen_11.1.13.0.exe 软件,然后就可以开始安装 Navicat 软件了。

(1) 双击安装文件 mysqlen_11.1.13.0.exe,进入安装欢迎界面如图 17-1 所示。单击“下一步”按钮,进入“许可证”界面,如图 17-2 所示。

(2) 选择单选按钮“我同意”,单击“下一步”按钮,进入“选择安装文件夹”界面,单击“浏览”按钮,可以选择合适的路径,如图 17-3 所示。



图 17-1　开始安装 Navicat 软件



图 17-2　安装许可界面

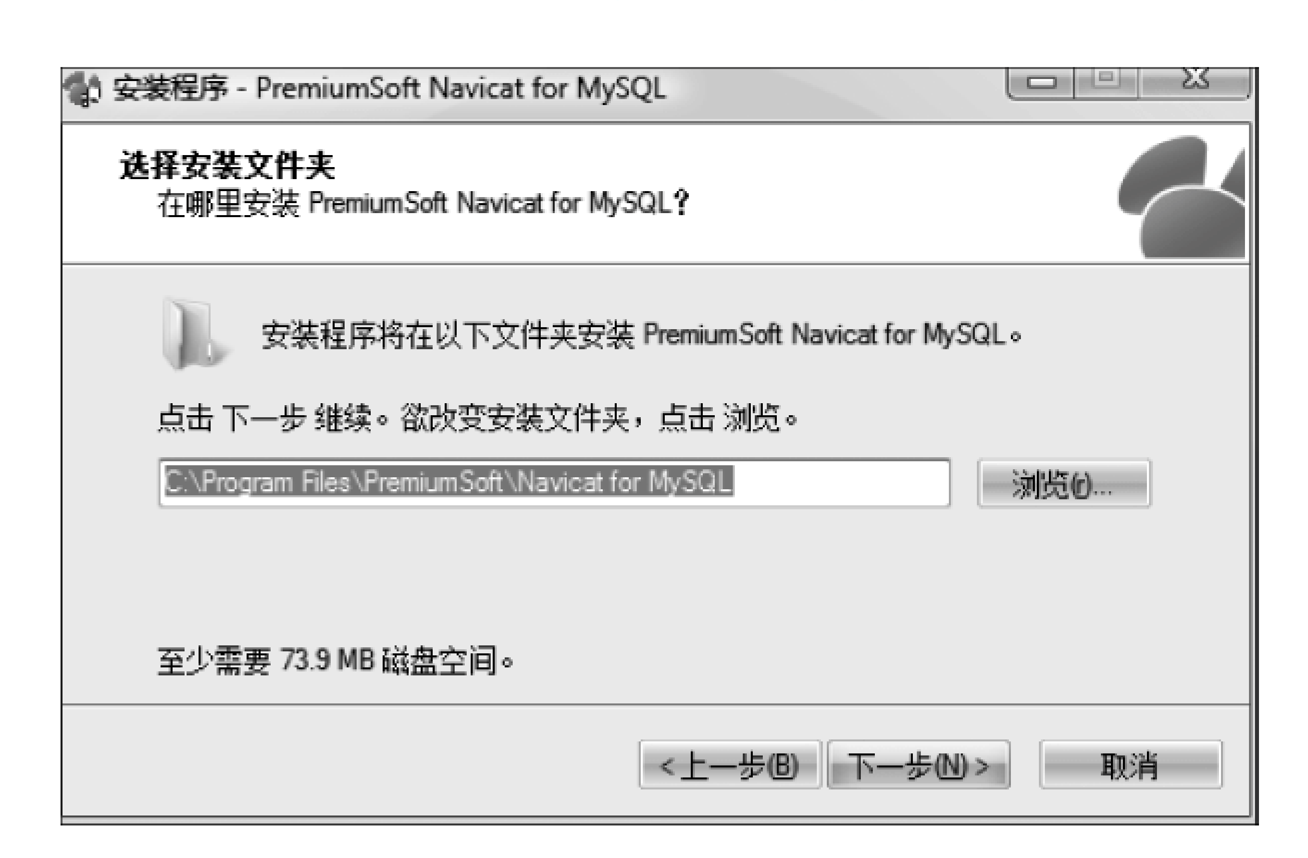


图 17-3　选择安装文件夹

(3) 单击“下一步”按钮，进入选择“选择开始目录”界面，依次单击“下一步”按钮，可以进入“准备安装”界面，如图 17-4 所示。

(4) 单击“下一步”按钮，进入“选择额外任务”界面，可以在此选择是否创建桌面图标，如图 17-5 所示。

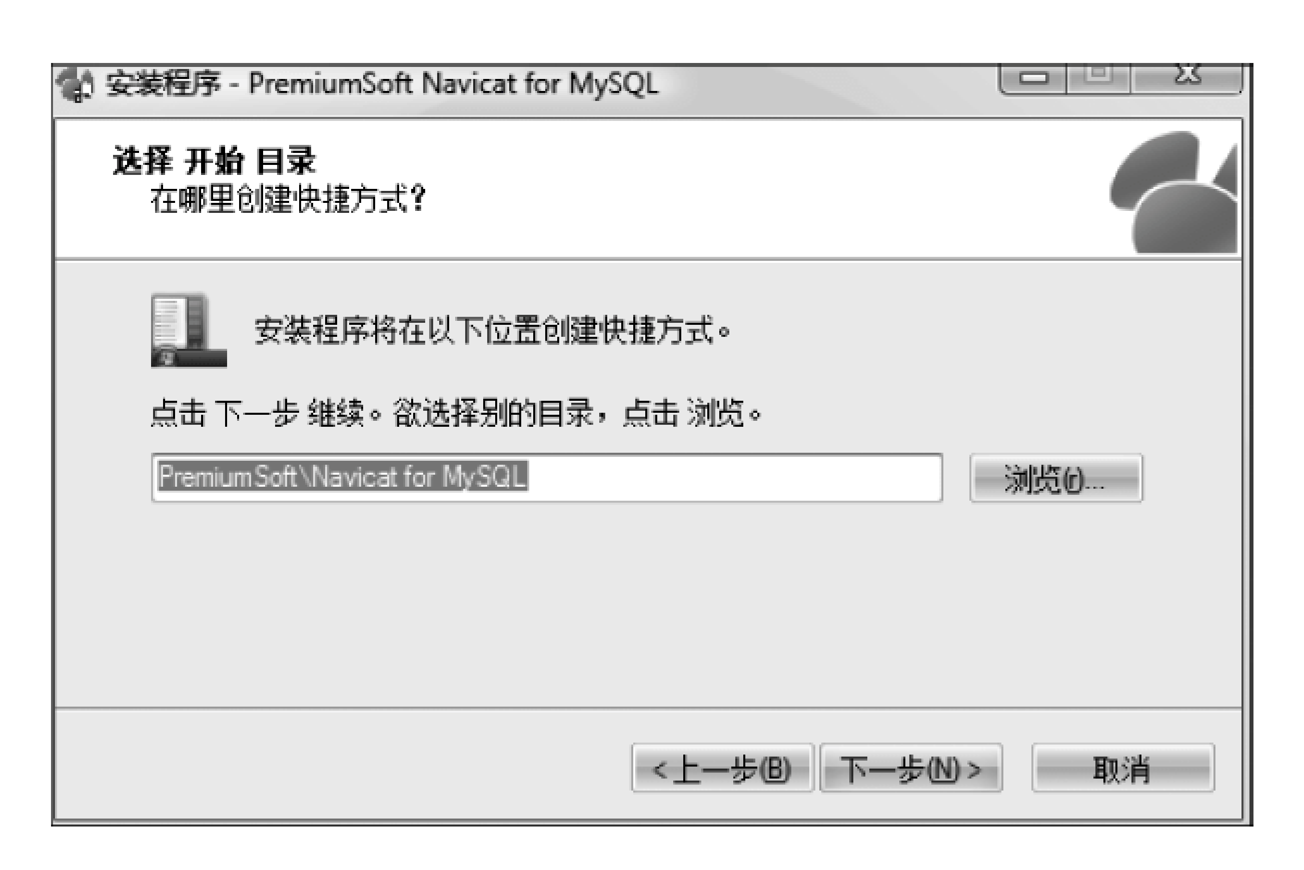


图 17-4　选择开始目录

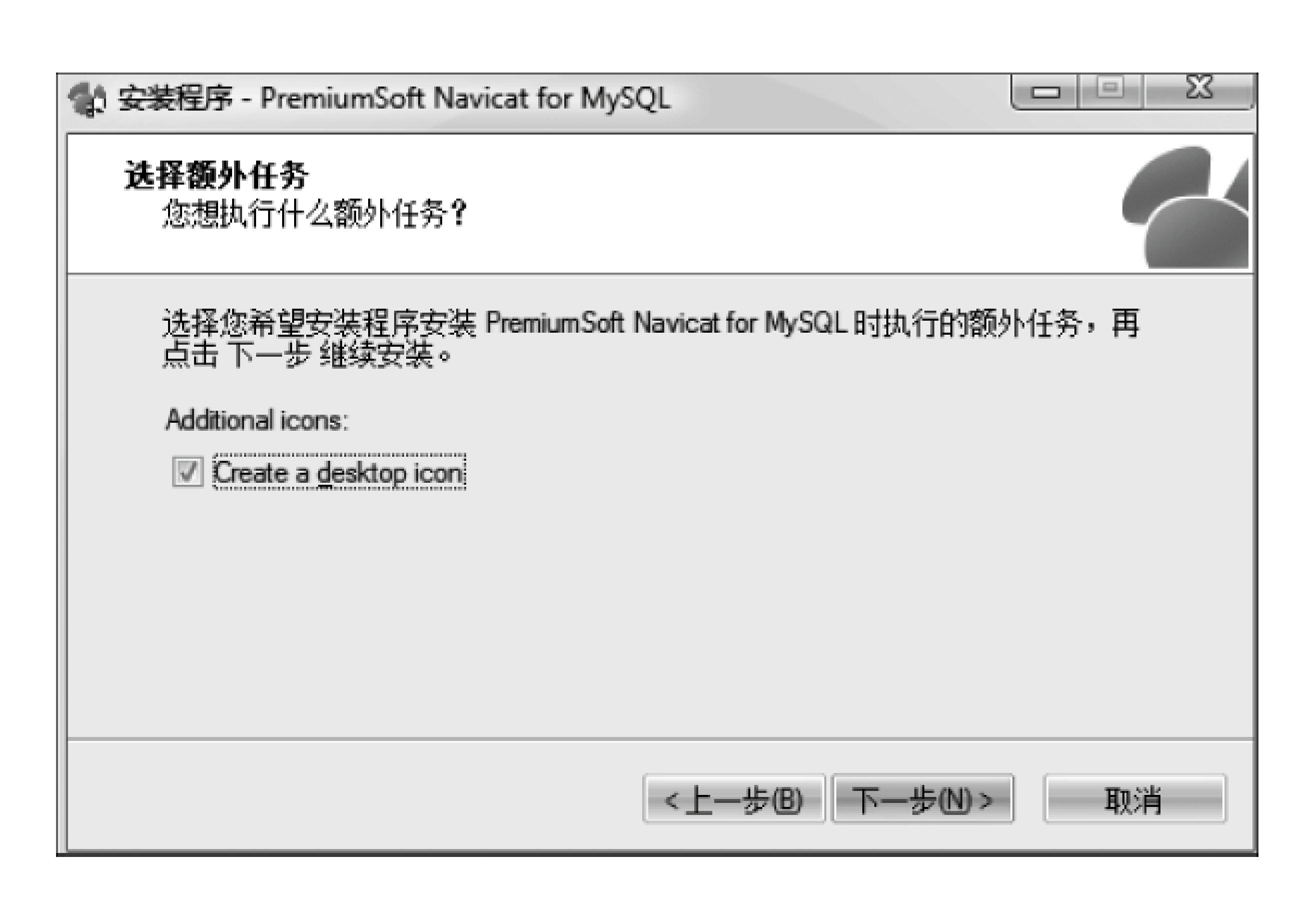


图 17-5　选择额外任务

(5) 单击“下一步”按钮，进入“准备安装”界面，如图 17-6 所示。单击“安装”按钮，进入安装过程。如图 17-7 所示，单击“完成”按钮，即可完成 Navicat 软件的安装。

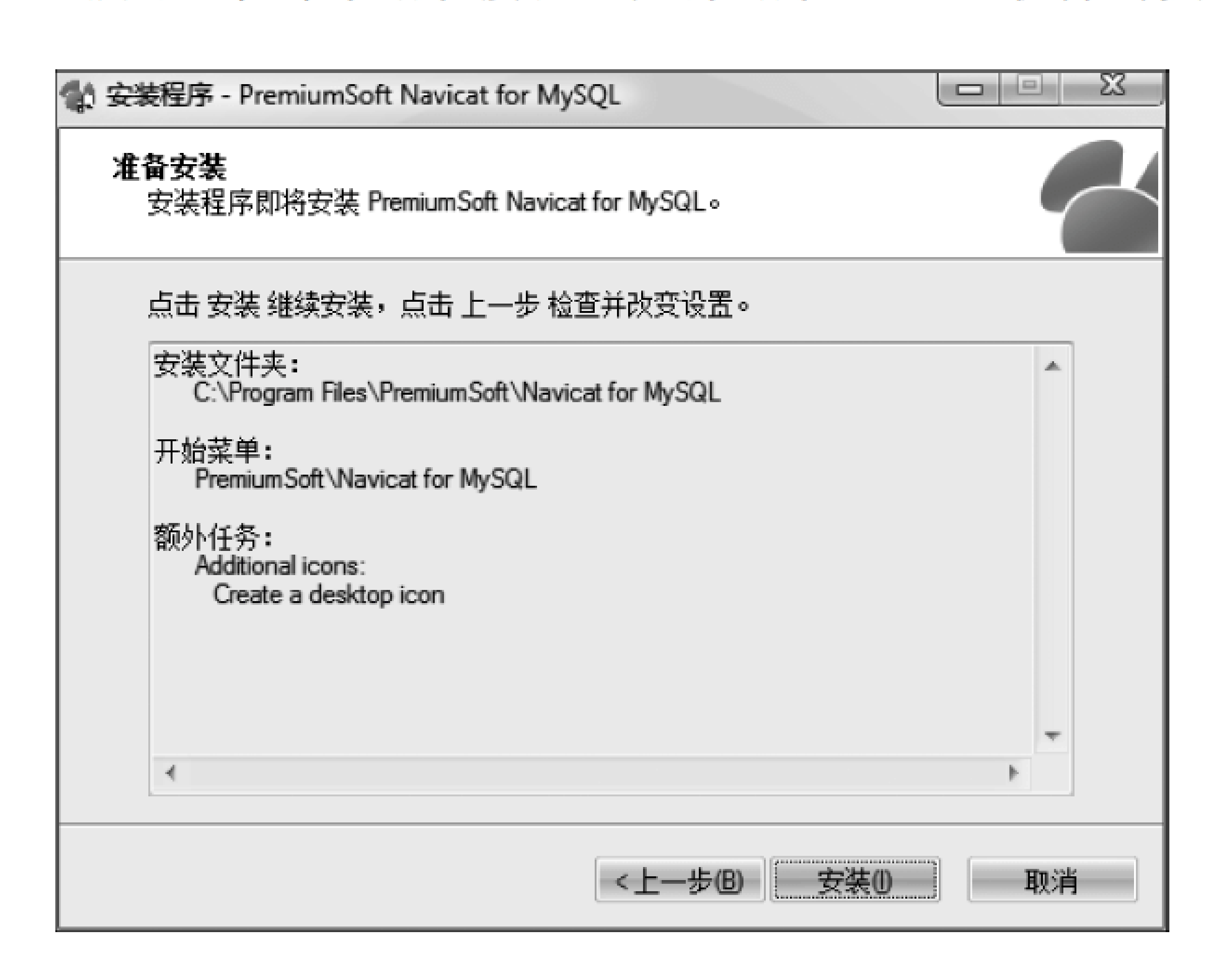


图 17-6　准备安装



图 17-7　完成安装

2. Navicat 的连接

(1) 选择"开始"→"所有程序"命令，按照如图 17-8 所示执行 Navicat for MySQL 命令，即可进入如图 17-9 所示的 Navicat 初始界面。

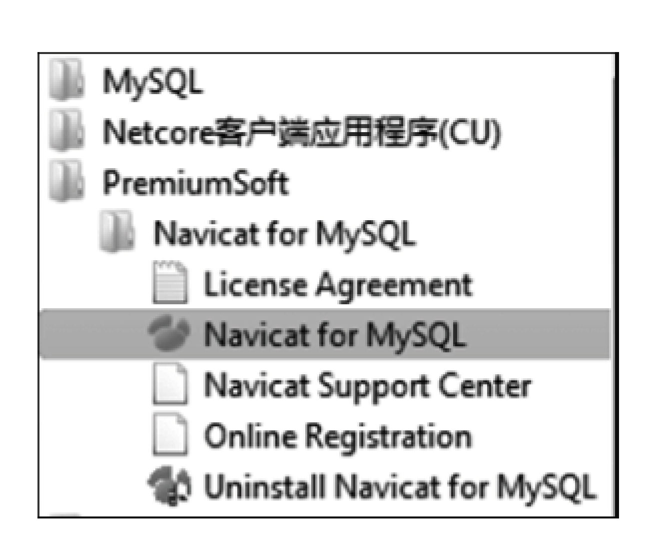


图 17-8　执行 Navicat for MySQL 命令



图 17-9　Navicat 的初始界面

(2) 在如图 17-9 所示的 Navicat 软件初始界面中，可以观察到 Navicat 软件的菜单栏、工具栏和连接两个操作窗口。菜单栏中包括"文件""查看""工具""窗口"和"帮助"常用操作命令菜单。工具栏中包括"连接""管理用户""表""视图""存储过程"和"事件"等数据库对象的常用操作提示。

(3) 单击工具栏中的"连接"按钮，进入如图 17-10 所示的"连接"界面。在如图 17-10 所

示的界面中输入连接名等选项，然后单击“连接测试”按钮，测试连接成功后就可以使用Navicat 软件实现 MySQL 数据库的可视化操作了。



图 17-10 “连接”界面

17.1.2 数据库管理的基本操作

1. 打开数据库

(1) 右击“连接”窗口的“连接”图标，如图 17-11 所示，在弹出的菜单中执行“打开数据库”命令，系统弹出如图 17-12 所示的“打开数据库”对话框。



图 17-11 选择要执行的操作

(2) 在如图 17-12 所示的“打开数据库”对话框中输入指定数据库 teaching，单击“确定”按钮，软件即可打开 teaching 数据库，即 teaching 变成当前数据库，在图中的连接窗口中呈现亮色。同时，工具栏中的相关数据库对象图标，如“表”“视图”“存储过程”“事件”“查询”和“备份”等，也变成可操作的状态了。



图 17-12　打开数据库 teaching

2. 表的管理

(1) 在“连接”窗口中，单击数据库 teaching 前面的“+”号，展开 teaching 数据库，单击“表”后，界面如图 17-13 所示。

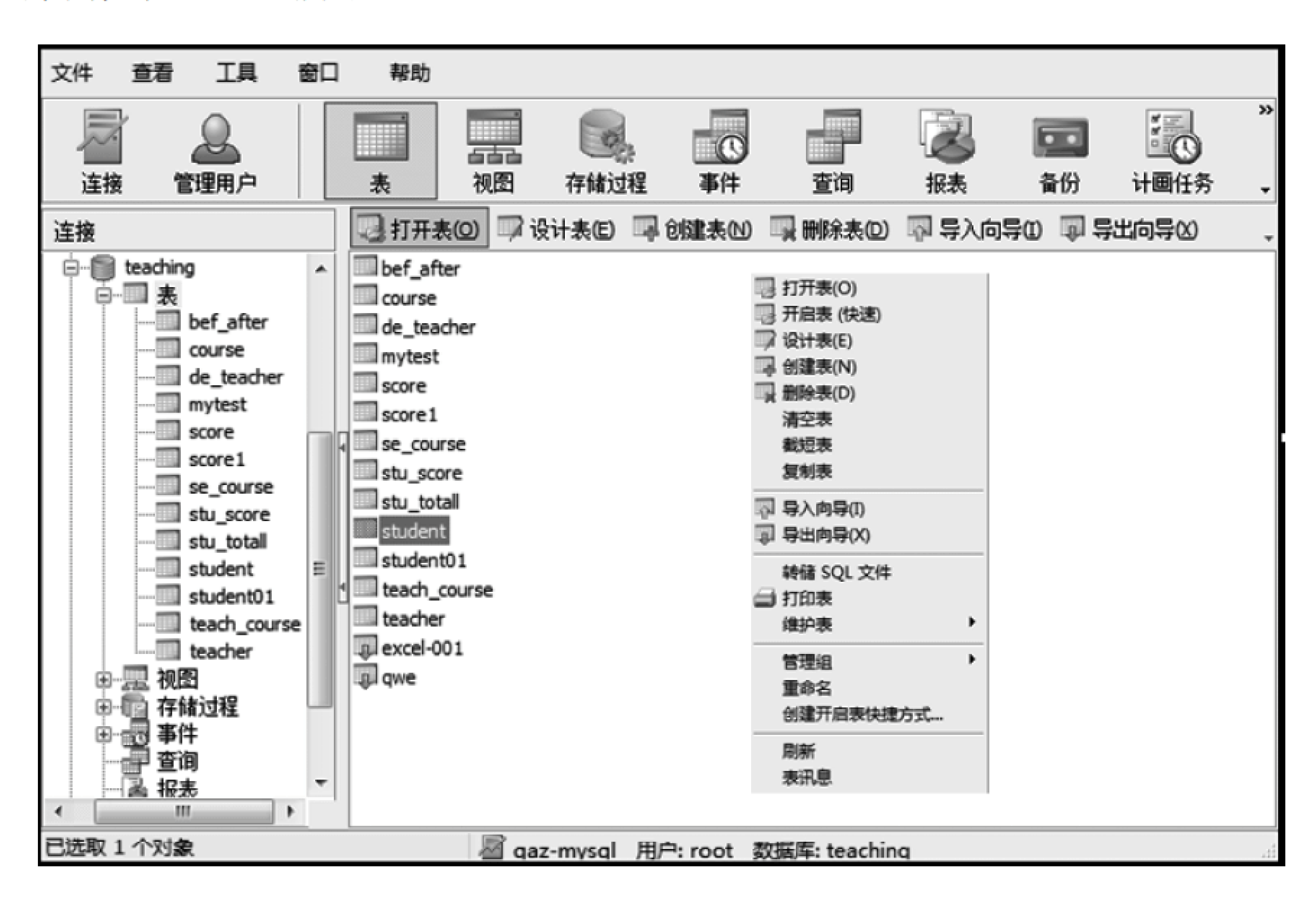


图 17-13　选择表

(2) 选择 student 表，单击工具栏中的“打开表”按钮，或者右击 student 表，在弹出的菜单中执行“打开表”命令，即可显示如图 17-14 所示的 student 表的浏览状态。

(3) 在此状态下，还可以对 student 表进行如导入、导出、创建视图等操作。

3. 管理存储过程

(1) 单击工具栏中的“存储过程”按钮，如图 17-15 所示，可以查看 teaching 数据库中的存储过程。

(2) 在此窗口中，可以实现创建、设计和删除存储过程的操作。可以通过工具栏按钮进行操作，也可以通过快捷菜单执行相关命令进行操作。

(3) 类似的如视图、事件、查询、报表和触发器等都可以在此窗口中进行管理操作。



studentno	sname	sex	birthdate	entrance	phone	Email
18122210009	许东山	男	1999-11-05	789	13623456778	qwe@163.com
18122221324	何白露	女	2000-12-04	879	13178978999	heyy@sina.com
18125111109	敬横江	男	2000-03-01	789	15678945623	jing@sina.com
18125121107	梁一苇	女	1999-09-03	777	132123456777	bing@126.com
18135222201	凌浩风	女	2001-10-06	867	15978945645	tang@163.com
18137221508	赵临江	男	2000-02-13	789	12367823453	ping@163.com
19111133071	崔依歌	女	2001-06-06	787	15556845645	cui@126.com
19112111208	韩山川	男	2001-02-14	666	15878945612	han@163.com
19122203567	封月明	女	2002-09-09	898	13245674564	feng@163.com
19123567897	孙释平	女	2002-08-04	999	13175689345	pingan@163.com
19126113307	梅惟江	女	2003-09-07	787	13245678543	zhu@163.com
1928261001	孙释远	女	1999-09-09	799	1398909876	sdy@163.com

图 17-14　浏览表

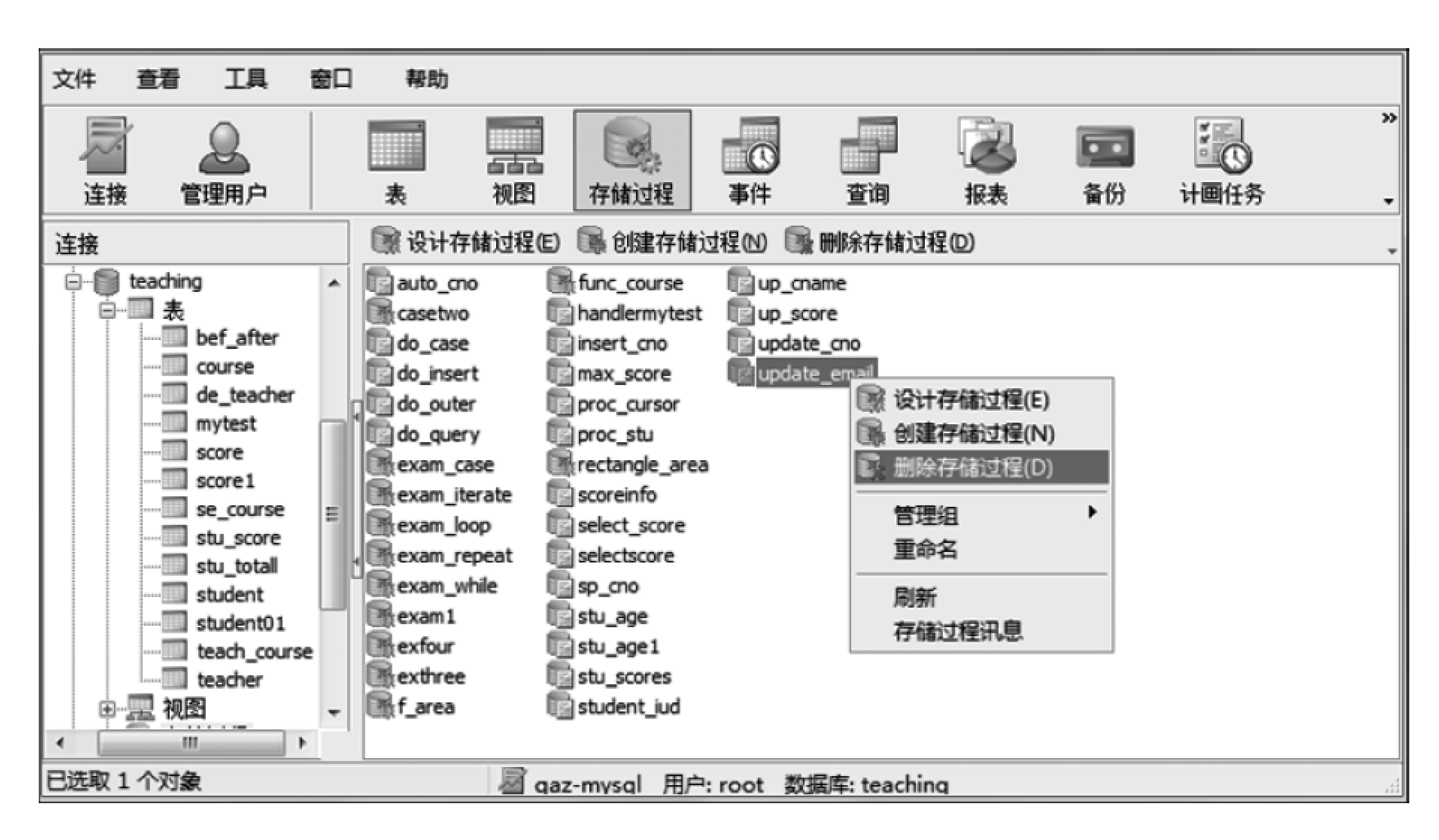


图 17-15　管理存储过程

4. 备份管理

(1) 单击图 17-15 所示的工具栏中的“备份”按钮，进入如图 17-16 所示的“创建备份”对话框。

(2) 单击“开始”按钮，系统就自动备份数据库 teaching，备份信息如图 17-17 所示。执行完毕保存备份即可。

5. 导入导出管理

(1) 数据的导入导出是针对表数据进行的操作。从打开数据库 teaching 的界面的工具栏中可以找到导入、导出按钮，也可以从图 17-13 中的右击菜单中执行导入导出命令。

(2) 执行“导出向导”命令，进入如图 17-18 所示的“导出向导”对话框，例如选择 txt 类型。



图 17-16　创建备份



图 17-17　执行备份



图 17-18　选择导出文件类型

(3) 单击“下一步”按钮，进入如图 17-19 所示的选择导出表和导出文件的对话框中。例如，分别选择 student、course 和 score 三个表，并指定导出文件的存储路径。如果单击“高级”按钮，可以为导出文件选择编码格式和时间戳记，如图 17-20 所示。



图 17-19 选择导出表



图 17-20 选择编码格式和时间戳记

(4) 单击“下一步”按钮，进入如图 17-21 所示的“选择导出列”对话框，在此可以对导出表进行选择全部或部分列的操作。



图 17-21 选择导出列

(5) 单击“下一步”按钮，进入如图 17-22 所示的“定义额外选项”对话框，在此可以选择是否“包含列标题”“遇到错误继续”以及定界符、日期分隔符等选项。



图 17-22　定义额外选项

(6) 单击“下一步”按钮，进入如图 17-23 所示的“信息收集完成”对话框。

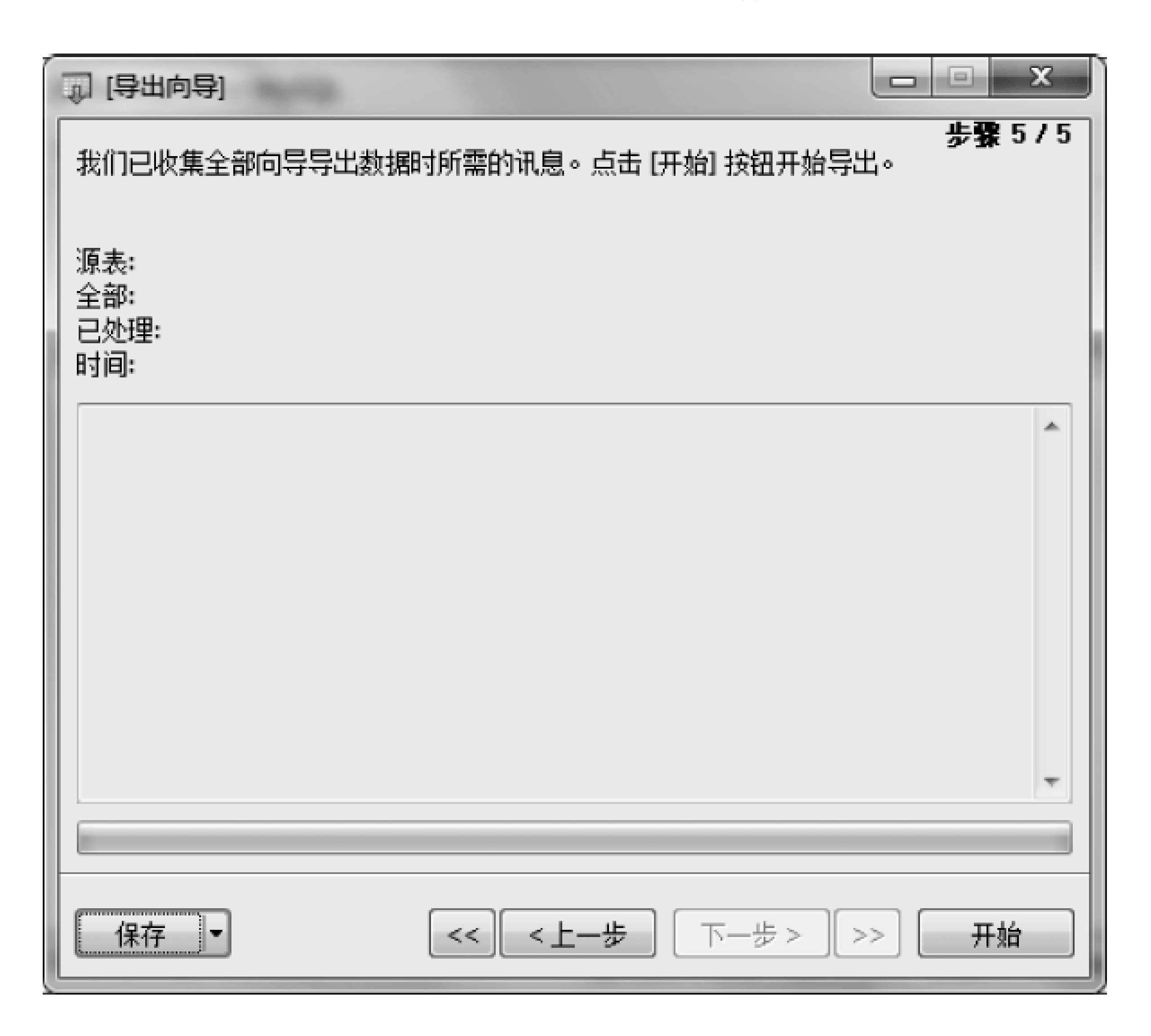


图 17-23　信息收集完成

(7) 单击“开始”按钮，即可完成选择数据表的导出操作，如图 17-24 所示。

(8) 如果想进行导入操作，可以选择图 17-13 所示的“导入向导”命令，进入如图 17-25 所示的“导入向导”对话框，按照向导说明，就能够实现将其他类型数据导入到当前数据库中的操作。导入过程与导出操作类似，在此不再赘述。

Navicat 软件是 MySQL 数据库的一个性能优良的可视化管理工具，功能强大，支持中文操作。在实验过程中利用 Navicat 软件操作 MySQL 数据库会收到很好的效果。



图 17-24　完成导出操作



图 17-25　导入向导

17.2　可视化工具 phpMyAdmin 软件

phpMyAdmin 是一个用 PHP 编写的，可以通过互联网控制和操作的 MySQL 数据库。通过 phpMyAdmin 完全可以对 MySQL 数据库进行操作，例如建立、复制和删除数据等。使用 Web 接口可以成为一个应用简易方式输出 MySQL 语法的较好途径，尤其是要处理海量信息的查询与分析更为方便。

17.2.1 phpMyAdmin 工具的安装和配置

phpMyAdmin 是最常用的 MySQL 维护工具，是用 PHP 开发的基于 Web 方式架构在网站主机上的 MySQL 管理工具，支持中文，管理数据库非常方便。可以在其官方网站 https://www.phpmyadmin.net/downloads/下载最新版本的 phpMyAdmin 工具。

(1) 下载 phpMyAdmin 软件。到官方网址下载 phpMyAdmin 软件，建议使用多国语言版，这样才会出现简体中文字符。例如，下载 phpMyAdmin-4.7.3-all-languages.zip 版本。

(2) 配置 phpMyAdmin。下载的 phpMyAdmin 解压，并将解压出来的文件夹改名为 phpMyAdmin，然后查找到此文件夹的/phpMyAdmin/libraries 文件夹下的 config.default.php，复制到/phpMyAdmin 文件夹，并重命名为 config.inc.php。

① 用文本方式打开 config.inc.php，执行“编辑”→“查找”命令。

查找下列文本：

```
$cfg['PmaAbsoluteUri'] = '';
```

将其改成

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin';
```

查找下列文本：

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

把 config 改成 http，即

```
$cfg['Servers'][$i]['auth_type'] = 'http';
                                   //改成 http 后，就可以像登录路由器那样，出现一个登录对话框
```

然后保存退出。

② 对于 Windows 系统：加载 mbstring 解决方法。利用文本方式 PHP 的配置文件 php.ini。

查找下列文本：

```
;extension = php_mbstring.dll
```

将其改成

```
extension = php_mbstring.dll
```

然后保存退出。

(3) phpMyAdmin 软件的运行。本地配置完毕后，将 phpMyAdmin 软件上传到服务器或虚拟服务器上的 Web 的根文件夹 htdocs(有些服务器文件夹 httproot)中。用浏览器输入地址，例如：http://localhost/phpmyadmin/(如果默认端口号 80 被占用，可以改成其他端口号，如 81。输入网址 http://localhost:81/phpmyadmin/即可)，就会跳出一个输入用户名和密码的对话框，如图 17-26 所示，输入 MySQL 的账号和密码即可开始利用 phpMyAdmin 软件操作 MySQL 数据库。



图 17-26　运行 phpMyAdmin 软件

17.2.2　phpMyAdmin 工具的常用操作

利用 phpMyAdmin 软件操作 MySQL 数据库，包括数据库、表、存储过程、函数程序、触发器和事件等对象的创建和使用，管理用户账户和权限，进行数据备份和数据恢复等，可以实现对 MySQL 数据库的快捷方便的操作管理。

1. 登录 phpMyAdmin 软件

（1）在登录 phpMyAdmin 的身份验证窗口中输入 MySQL 的账号和密码，如图 17-27 所示，单击"登录"按钮，进入如图 17-28 所示的初始界面。

（2）在如图 17-28 所示的页面中，用户修改密码、选择服务器字符排序规则和语言等，也可以操作数据库、执行 SQL 语句、查看运行状态、管理账户，进行数据的导入导出，以及查看字符集、变量、二进制日志、引擎和插件等。

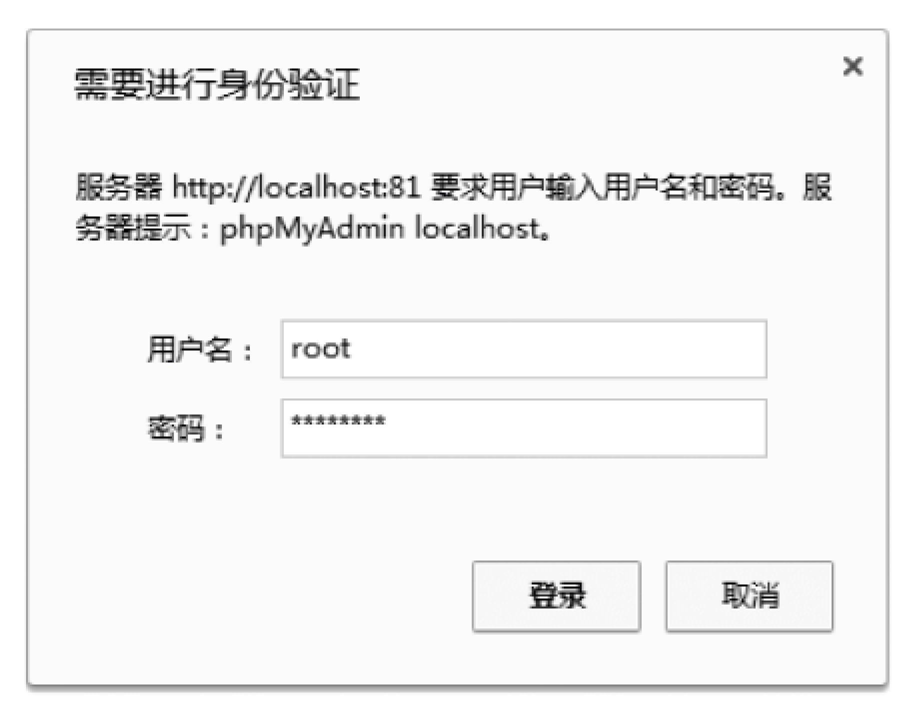


图 17-27　登录 phpMyAdmin 软件



图 17-28　phpMyAdmin 软件的初始界面

2. 管理数据库对象

(1) 数据表管理。如图 17-28 所示，展开“表收藏夹”下的 teaching 数据库，单击 teaching 数据库的“表”，进入如图 17-29 所示的表的操作窗体中，可以看到该数据库下的所有表。此时，可以对每一个表进行浏览数据、修改结构、搜索记录、插入记录、清空记录、删除记录等操作，也可以单击“表”文件夹下的“新建”命令创建新表。

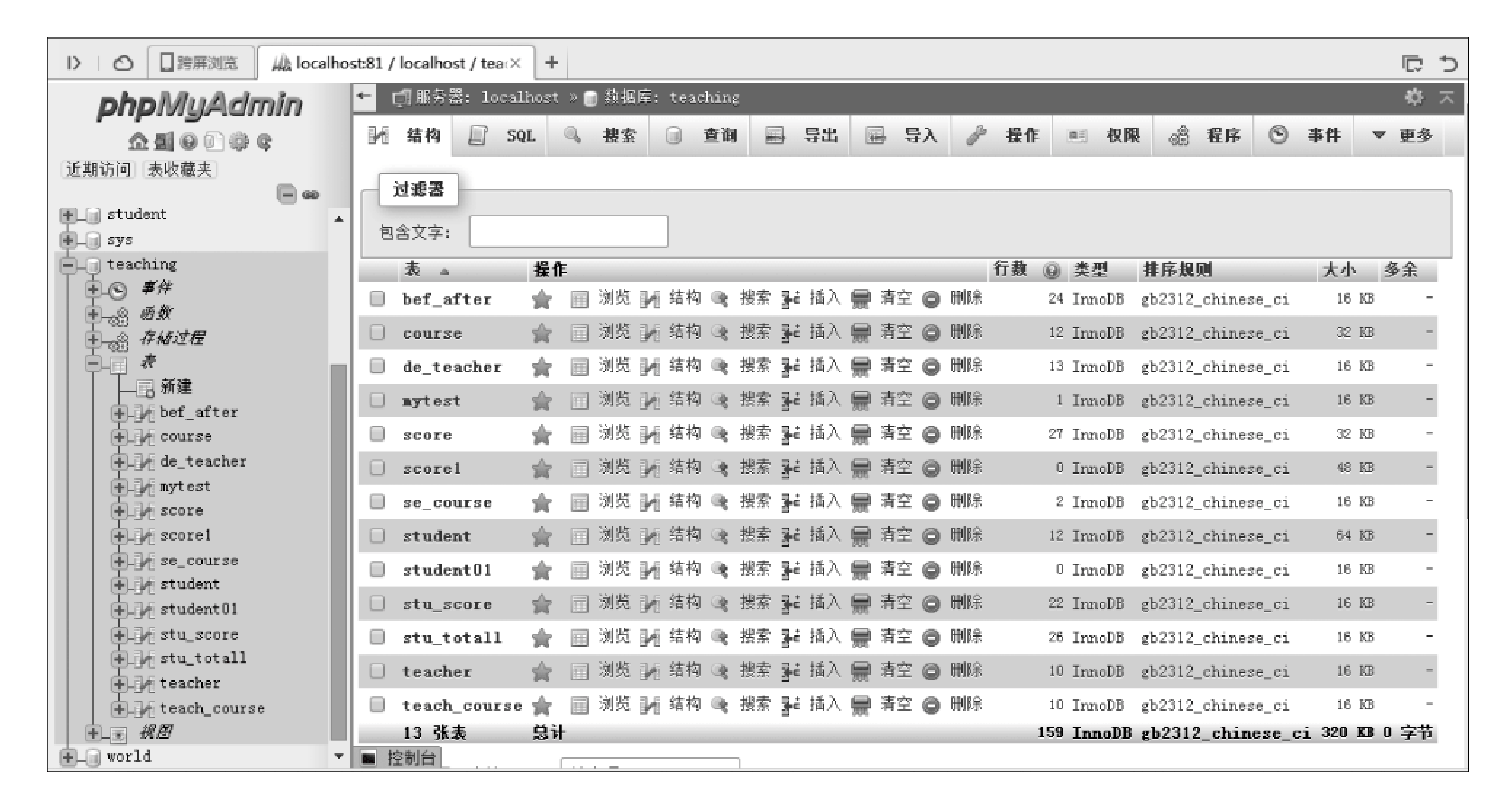


图 17-29 表的管理

(2) 表结构的管理。单击一个表，如 course，即可进入如图 17-30 所示的表结构修改页面。此时，可以对 course 表中的列进行编辑修改、复制和删除等操作。



图 17-30 表结构修改

(3) 存储过程管理。单击 teaching 数据库下的“存储过程”，展开存储过程列表，可以对存储过程进行操作。单击存储过程 auto_cno，如图 17-31 所示，可以修改存储过程 auto_cno 的所有选项参数，如存储过程名、代码等内容。



图 17-31　存储过程管理

3. 设计器的使用

使用 phpMyAdmin 软件可以实现 MySQL 的一般操作：如导入导出数据、创建索引和实现参照完整性约束、使用查询等，还能够管理视图、存储过程、存储函数、触发器、事件、用户、权限，以及备份与恢复数据库等。

另外，phpMyAdmin 软件的设计器的功能也很强大。

（1）选择数据库 teaching，在工具栏中单击“设计器”按钮，窗体中就会显示数据库 teaching 中的所有表和视图的结构，如图 17-32 所示。



图 17-32　利用设计器管理数据库

(2) 在图 17-32 的右侧窗口中，可以观察到工具栏，展开工具栏，如图 17-33 所示，通过该工具栏，可以对当前窗口中的表、视图等进行操作。

(3) 执行工具栏的第一条"show/hide tables list"命令，可以在窗体右侧显示数据库 teaching 中的所有表和视图，如图 17-34 所示，在此可以通过列表前的选择按钮选择表结构是否在窗体中显示，以此可以在表设计器中实现对表和视图的相关操作。



图 17-33 展开工具栏

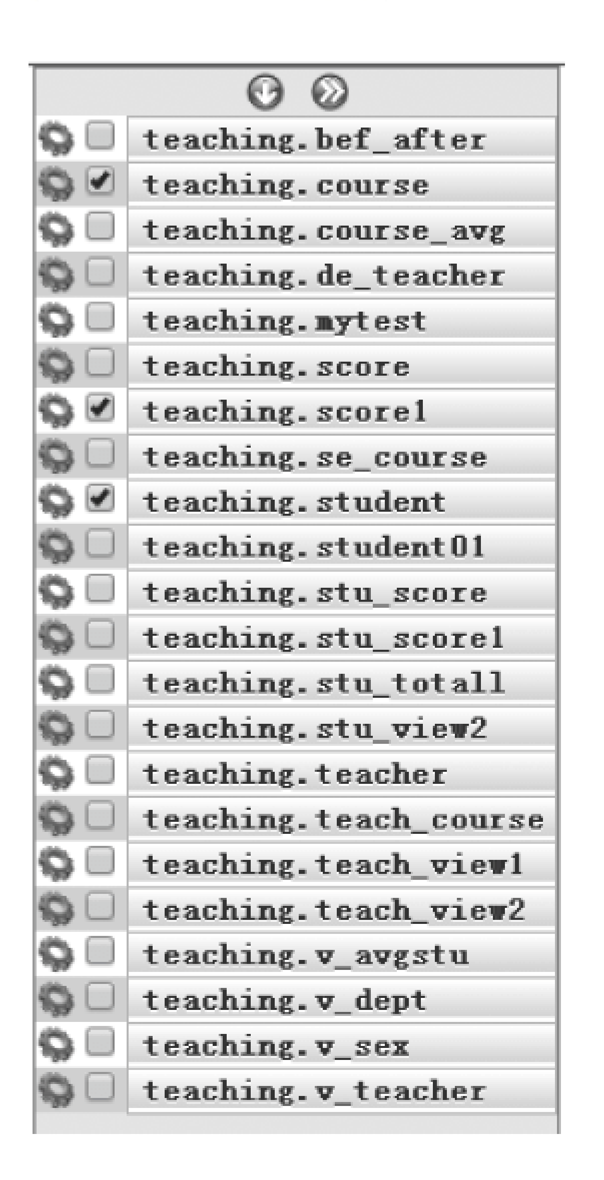


图 17-34 数据表和视图

MySQL Workbench、phpMyAdmin 和 Navicat 软件是目前在操作 MySQL 数据库时常用的可视化操作工具，在实验过程中，可以选择符合环境要求的软件使用。其中 phpMyAdmin 软件需要在 PHP 环境下运行和操作。

第18章　MySQL 5.7 实验指导和实验内容

学习 MySQL 数据库需要在掌握部分理论内容的同时进入实验环节。只有在熟悉 MySQL 的运行环境的基础上进行实验操作，才能实现培养学生应用创新能力，以提高学生的数据库设计、管理和操作的水平。

本课程实验的学时数可安排 32 学时左右，共安排 16 个实验。任课教师可根据实际需要对实验内容进行摘选或补充。讲课与上机的课时比例应该控制在 3∶2 左右。

18.1　MySQL 5.7 实验指导

1. 上机实验的目的和要求

学习 MySQL 数据库应当在达到能够独立地创建和组织数据库、独立上机调试程序和分析结果的目标，进一步加深对 MySQL 数据库的有关理论的理解及编程的基本技能的提高。所有计算机类课程都要求实践性经验，没有足够的上机实践，学好 MySQL 数据库是不可能的。

上机实验的目的不仅仅是为了验证教材和讲课的内容或者验证自己所编程序的正确与否，更重要的是通过实际题目的上机实验，可以提高动手能力、提高分析问题和解决问题的能力，最终能够达到举一反三的自主学习效果，以至于将来能够在实际的软件开发过程中具有创新能力。

具体来说，上机实验的目的应该从以下几个方面进行理解。

(1) 加深对 MySQL 语言基础的理解。对于初学者来说，仅仅通过课堂教学，很难将课程要求的内容记住并理解透彻，尤其是比较琐碎的语言要素，如数据类型、运算符、系统函数和灵活多变的规则等。只有通过多次上机，在实验中不断探索、思考、归纳和总结，才能够熟练地掌握相关内容。通过实验可以快速地掌握语法规则、验证自己的能力、加深自己对数据库理论的认识。

(2) 了解和熟悉 MySQL 语言的环境。一个程序必须在一定的硬件和软件环境下才能运行。使用者应该了解，为了运行一个 MySQL 程序需要哪些必要的外部条件(例如硬件配置、软件配置)，可以利用哪些软件的功能来帮助自己开发数据库。MySQL 中几乎所有的基本操作都可以通过可视化工具来实现。MySQL 用户的数据文件、SQL 语言文件的存储也应该设置固定的路径，防止出现文件存放位置混乱，在调试程序时浪费较多的资源。同时要注意保存现有的数据，以备下次实验使用。

(3) 实验模式的探讨。MySQL 的上机实验贯穿整个数据库操作的教学过程。实验不再是单一的上机调试程序，建议根据学生的基础和课时要求按阶段分成两种模式：上机验

证和自由上机。

上机验证是指学生根据实验指导书的操作步骤进行操作，并认真做好记录。这种方式一定让学生先做好实验的预习，教师在答疑过程中，对于普遍存在的问题及时进行总结。

自由上机是指当学生掌握了调试程序的基本方法后，可以采取自由上机的方式。目前许多学校提供电源和网络环境，让学生能够利用自己的笔记本电脑进行自由上机。这样一来，学生就可以自己安排时间进行试验，比利用公共计算机省去了环境配置和数据管理等基本操作的时间，从而进一步提高学生调试程序的效率和能力，增强学生学习数据库语言的自信心。

(4) 充分利用网络资源。建立班级或课程交流平台，把各种网络教学资源有机地整合到实验教学中。对于实验过程中遇到的问题，可以通过网络搜索，集群策群力来解决问题，改善学生的学习习惯，提高学生学习的主动性。

2. 实验内容的设置

针对本课程的实验内容的不同特点，可以将实验分为熟悉环境阶段、基本操作阶段和技术提高阶段共三个阶段，其主要内容和基本要求如下。

(1) 熟悉环境阶段。前期准备阶段的主要任务是理解数据库、数据模型和数据库系统的基本概念；根据要求自己设计出一个结构合理的数据库，并且熟悉 MySQL 集成环境的使用和数据库及表的建立，数据库完整性设置等。该阶段可以针对教材的第 1 章和第 2 章内容进行设置。

(2) 基本操作阶段。MySQL 实验的第 2 阶段为基本操作阶段。主要任务是掌握 MySQL 语言的基本操作，包括 SQL 查询语句以及与 MySQL 语言相关的数据库、表、视图、索引、存储过程、存储函数、游标、触发器、事务和锁的基本操作，并能够针对各种数据库对象在实际中的简单应用进行了解。该阶段可以针对教材的第 3～9 章内容进行设置。

(3) 提高阶段。技术提高阶段的实验，要求学生在掌握课本内容的基础上，自学一些相关知识，技术提高阶段的主要任务是掌握数据库系统的设计与实现的知识内容，将理论知识与实际问题相结合。该阶段可以针对教材的第 10～15 章内容进行设置。这些内容的相关软件包括 PHP、JSP、IIS、Apache、tomcat 以及集成开发环境 IDE 软件 NetBeans 等，这些都是目前软件开发企业正在应用的软件。熟悉这些软件的环境和操作对于学生的将来工作提供了良好的先机。

3. 教学重点与难点的处理

(1) 客户端工具。学会在 DOS 窗口进行各种 MySQL 语句的操作格式、过程和技巧，这是 MySQL 实验的最常用方式。也要学会一些其他操作，例如选择、复制、粘贴等，以及在此操作环境下的一些快捷键的操作。

(2) 数据库和表。数据库是存储和管理 MySQL 数据的模式。要对数据库引擎做一些简单的了解，特别是 InnoDB 的特点。

(3) MySQL 语言。MySQL 语言的组成要素多，规则繁杂，应用广泛，非常容易出错。一些常用的关键词、常量、变量、函数、运算符和 SQL 语句都需要下一些工夫去熟练掌握，就像学习化学化工类课程，化学元素符号和常用元素特性必须记牢，学习物理要记住一些常用的公式和参数一样，只靠理解是远远不够的。select 语句对初学者来说是比较复杂的，本课程先介绍基本的内容，即分清楚 select、from、where、order、group by、having、limit 等子句的

作用，再介绍复杂的多表连接和子查询操作，然后学习与 MySQL 语句相关的内容。

(4) 常用数据库对象。常用数据库对象包括索引、视图、存储过程、存储函数、触发器、事件、游标等。主要是掌握格式定义、编写程序和应用过程的操作。MySQL 中的存储程序本身能够实现许多重要功能，同时也是触发器、事件、游标和控制流语句的应用载体和中介。

(5) 备份和恢复。计算机用户能够对一些重要文件、资料定期进行备份是必要的。通过恢复启用原先的文件备份，就可以节省大量的资源。

(6) 权限管理。数据库用户访问到数据库中的数据对象，是根据管理员分配给他的权限决定的。不同的用户具有不同的访问权限，常见的访问权限有 select 查询权限、update 更新权限、insert 录入权限和 delete 删除权限。

(7) 日志和性能优化。MySQL 的日志管理使用方便，功能强大，是管理 MySQL 服务器和客户端的重要工具。MySQL 的性能优化对于管理大数据，特别是优化查询操作，起到了不可替代的作用，这也是 MySQL 的特色。

(8) 可视化工具的使用。可视化工具的灵活运用是学习 MySQL 的基本操作的有力补充。无论是教材中介绍的 MySQL 自带的可视化工具 Workbeach，还是本书第 17 章介绍的 Navicat 和 phpMyAdmin 软件，都具有很强大的功能。只要熟悉一个可视化软件就可以从不同的视角对 MySQL 中的数据库对象和基本操作语句进行更深入的理解。尤其是熟悉软件中的一些图形向导、查询设计、工具栏、菜单等操作，会把使用者学习 MySQL 的兴趣提高到一个新的高度。在实验过程中，有许多可视化操作和命令操作在本质上是相通的，实验过程中应该多总结和归纳。

实际操作和反复探索是掌握 MySQL 数据库管理技术的必经之路，也是理解和掌握本课程难点和重点、提高自己创新能力的重要途径。

4. 上机实验前的准备工作

在上机实验前的准备工作做得好，可以使得上机实验得到事半功倍的效果。准备工作至少应包括：

(1) 第一次实验前，教师需要向学生说明实验的整体要求及实验的目标任务。说明实验安排和进度、实验守则及实验室安全制度。

(2) 开始做实验就要介绍上机操作的基本步骤，注意了解 MySQL 实验过程中常见的错误，要学会使用网络搜索功能。

(3) MySQL 是理论与实验相结合的课程。上机前应当先了解相关的理论知识，再预习实验内容、方法和步骤，避免出现边操作边熟悉实验内容的现象。

(4) 该课程实验是连续的整体，需要有延续性。机房应有安全措施或学生自己配备一些常用的存储设备。避免前面的实验数据、程序被清除、改动影响后面的实验操作效率。

(5) 课下多做操作练习，如果能结合实际课题进行训练，会达到更好的效果。

教师可以在本实验内容完成的基础上，再根据学生的掌握情况和上机时间的长短指定课后习题或作业题作为实验内容。

本书给出 16 个实验内容，每一次实验包括 4 个以上的操作模块，上机时间每次为 2 小时左右。实验中用到的数据，基本上都是采用 teaching 数据库中的相关数据。

5. 学生在实验过程中应达到的基本要求

一个完整的 MySQL 实验的完成，除了在掌握必要的理论的基础上进行认真的操作以

外，还要记录好必要的实验结果，养成良好的实验习惯，进而完成实验报告。

(1) 实验之前对本实验对应的理论内容进行预习。

(2) 实验过程中要遵守实验室纪律，不做与本实验无关的事情。

(3) 实验过程中应及时发现问题并解决。

(4) 实验报告要求认真按规定格式书写，实验报告要求独立完成并按时上交。

学习计算机语言应该清楚一个这样的事实：教材不可能将软件的内容完全介绍，教师上课也不可能将所有的操作全部演示。MySQL 中的每一个数据库对象，都可以通过命令实行若干操作，也可以利用可视化操作形式实现。只要在掌握命令格式和功能的基础上，做一些验证实验，总结出该操作的主要内涵即可。

6. 实验报告的撰写

根据实验内容的记录整理出实验报告是实验过程中的一个重要组成部分。实验报告的格式要求规范、简洁、言之有物，以此达到锻炼学生的文字表达能力、提出和发现问题的能力、归纳和总结的能力的目的。撰写实验报告，切忌养成不加思考随意抄袭、拼凑一些可有可无的内容、编造实验结果记录等坏习惯。

实验报告一般应包括以下内容：

(1) 实验项目。如：实验 1 认识 MySQL 的运行环境。

(2) 实验目的。简述本次实验应该达到的目标。

(3) 实验内容。操作步骤、命令或程序清单、中间结果等。

(4) 运行结果。

(5) 总结本次实验过程中所取得的经验，分析实验过程中的出错原因并提出可能的解决方法。

18.2 MySQL 5.7 实验内容

实验 1 MySQL 5.7 的运行环境

1. 实验目的

(1) 熟悉 MySQL 5.7 的服务管理操作。

(2) 掌握 MySQL 5.7 的登录和退出操作。

(3) 掌握常用系统函数、运算符和表达式的功能和应用。

2. 实验预习与准备

预习主教材第 1 章和第 2 章：

(1) MySQL 5.7 环境的常用工具。

(2) 利用 select 语句进行输出的操作。

(3) MySQL 的服务器管理。

3. 实验内容及步骤

(1) MySQL 的启动和退出。

① 执行 cmd 命令启动。执行“开始”→“运行”命令，打开 DOS 窗口，在 DOS 提示符下输入登录命令后，按照指示位置输入密码，登录成功的结果如下：

```
C:\Users\Administrator>mysql -u root -p
Enter password: ******
mysql>
```

② MySQL 客户端登录数据库。在 Windows 7 操作系统中，执行“开始”→“所有程序”→ MySQL→MySQL 5.7 命令，该目录下包含 MySQL Command Line Client 和 MySQL Command Line Client-Unicode 两个选项。它们都是 MySQL 客户端的命令行工具。执行该命令后，按照提示位置输入密码，即可执行相关 MySQL 命令。

③ 退出。在“mysql>”提示符下，输入 quit 或 exit 命令，都可以退出 MySQL 登录。

（2）MySQL 服务的启动和退出。

① 执行“开始”→“运行”命令并输入 services.msc 后按 Enter 键也可以弹出图 18-1 所示的窗口。右击 MySQL 57 服务，可以在弹出的菜单中执行“启动”“停止”“暂停”和“恢复”等命令。

② 执行“开始”→“设置”→“控制面板”→“管理工具”→“服务”命令进行设置，打开的窗口如图 18-1 所示。右击 MySQL 57 服务，可以在弹出的菜单中执行“启动”“停止”“暂停”和“恢复”等命令。

③ 执行 cmd 命令启动服务。在 DOS 窗口中，net start mysql57 表示启动 MySQL 服务，net stop mysql57 表示关闭 MySQL 服务。

④ 选中图 18-1 的 MySQL57 服务项右击，在弹出的快捷菜单中选择“属性”命令，弹出如图 18-2 所示的对话框。



图 18-1 “服务”窗口

（3）常量和变量的应用。

① 查看字符“c”开头的系统变量命令。

```
mysql> show variables like 'c%';
```

② 查看当前 MySQL 软件的版本号和当前时间。

```
mysql> select @@version,current_date;
```

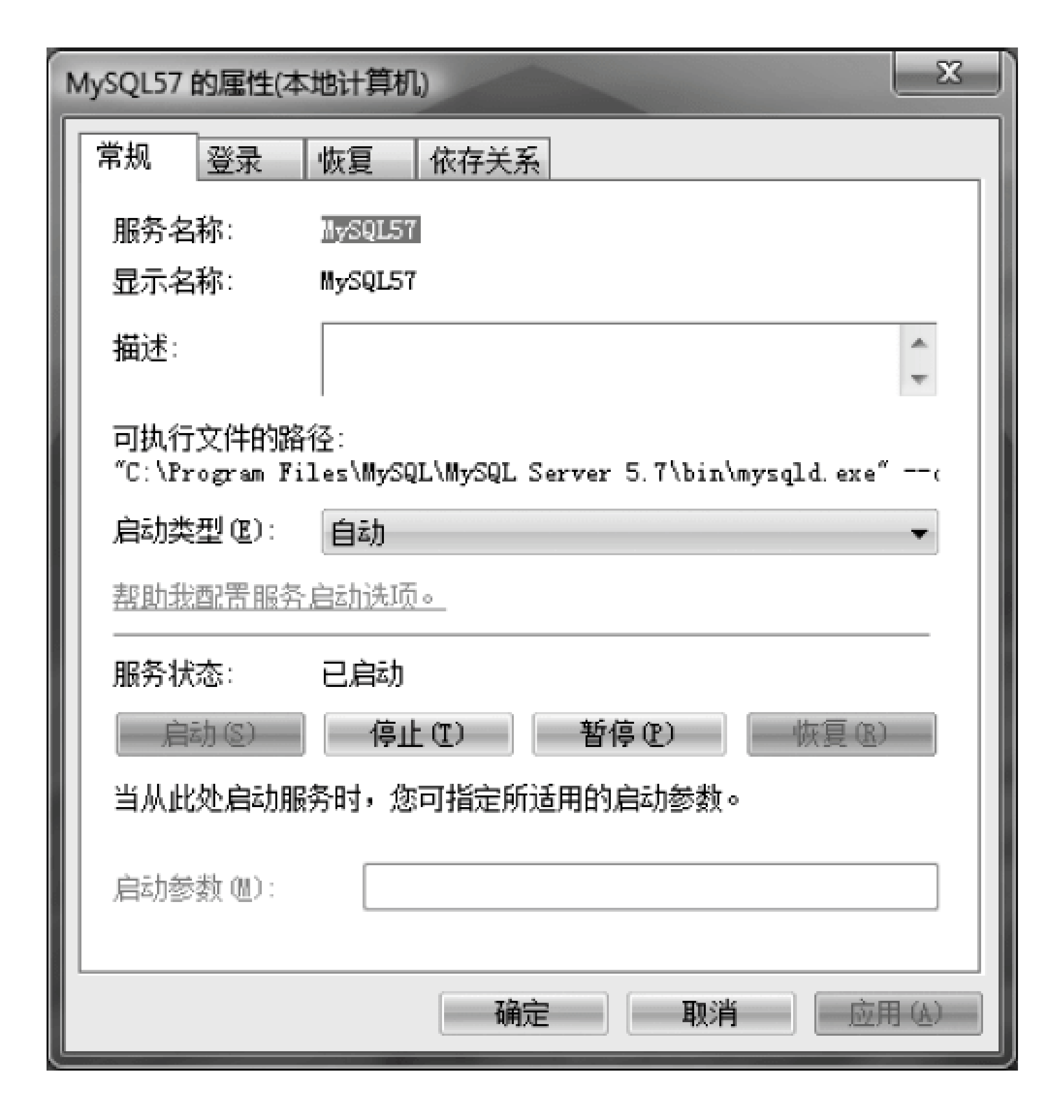


图 18-2　更改服务的启动类型

③ 针对当前会话，把系统变量 sql_select_limit 的值设置为 100。

```
mysql> set @@session.sql_select_limit = 100;
```

(4) 运算符的应用。

① 算术运算符：

```
mysql> select 5 + 2.7,1.5 * 3,3/5,10 - 27.5,5 % 3;
```

② 比较运算符：

```
mysql> select 'T'>'B',1 + 3 = 2, 'X'<'x', 8 <> 8, 'a'<= 'u';
mysql> select  null is  null,1.3 is null, 11.9 is not null;
mysql> select  'MySQL' like 'MY % ', 'APPLE' like 'AB_'
```

③ 逻辑运算符：

```
mysql> select ('T' = 'B'),('c' = 'C')and('c'<'D')or(7 = 8);
mysql> select ('A' = 'a') && (1 = 2),('F' = 'f') xor (1 + 1 = 2);
```

④ 位运算符：

```
mysql> select 5&1,7|3,1000 >> 5, ~9,6 ^7;
mysql> select 5&5,7|9,10 << 2, ~0,6 ^0;
```

(5) 系统函数的应用。

① 数学函数：

```
mysql> select cos(3.67),tan(4.71), sin(5),PI();
mysql> select  6 + round(10 * rand(),0) , 20 +  round(10 * rand(),0);
```

② 时间日期函数：

```
mysql> select dayofyear(20180515),dayofmonth('2018 - 05 - 15');
```

```
mysql> select dayofmonth('2018-07-15'),dayofweek(now());
mysql> select hour(curtime()),minute(curtime()),second(curtime());
mysql> select date_sub('2018-10-1 10:10:10', interval 27 minute);
```

③ 字符串函数：

```
mysql> select concat('you','are','best'),concat('123','456',null);
mysql> select right('ABCDEFGH',5),reverse(left('ABCDEFGH',5));
mysql> select length('12345abcde'),char_length('123 数据库技术 cde');
```

④ 其他函数：

```
mysql> select  version(),database(),user(),CONNECTION_ID();
mysql> select password('teach1727');
mysql> select format(7/3,3), format(123456.78,1);
```

4. 实验报告总结

(1) select 语句的输出功能。

(2) 常规标识符的命名规则。

(3) 常用运算符的使用方法及其优先级。

(4) 常用系统函数的功能和调用方法。

实验 2 数据库的基本操作

1. 实验目的

(1) 掌握数据库的存储方式。

(2) 掌握创建、修改数据库的方法。

(3) 掌握管理数据库的方法。

2. 实验预习与准备

预习主教材第 3 章：

(1) 数据库的基本概念和存储方式。

(2) 利用 MySQL 语句创建和管理数据库。

(3) 利用 MySQL 可视化工具管理数据库。

3. 实验内容及步骤

(1) 查看数据库。

① 查看 MySQL 数据库的存储位置。

参考安装路径：F:\Documents and Settings\All Users\MySQL\MySQL Server 5.7\Data。

② 查看当前系统中的数据库：

```
mysql> show  databases;
```

③ 查看系统中的数据库引擎：

```
mysql> show engines;
```

④ 设置 InnoDB 为当前数据库引擎：

```
mysql> set default_storage_engine = InnoDB;
```

(2) 数据库的基本操作。

① 创建 info1 数据库和 info2 数据库：

```
mysql > create database info1
    -> default character set gb2312
    -> default collate gb2312_chinese_ci;
mysql > create database if not exists info2;
```

② 练习从当前安装路径查看已经存在的数据库。

③ 打开数据库 info1：

```
mysql > use info1;
```

④ 修改数据库 info2：

```
mysql > alter database info2
    -> default character set gb2312
    -> collate gb2312_chinese_ci;
```

⑤ 显示数据库 info1 的结构信息：

```
mysql > show create database info1;
```

⑥ 删除数据库 info2：

```
mysql > drop database  info2;
```

(3) 利用 MySQL Workbench 创建和管理数据库。

① 从"开始"→"所有程序"中找到 MySQL 文件夹，执行 MySQL Workbench 命令，进入 MySQL Workbench 主界面，连接 MySQL 57。进入 MySQL Workbench 数据库操作的主界面，如图 18-3 所示。

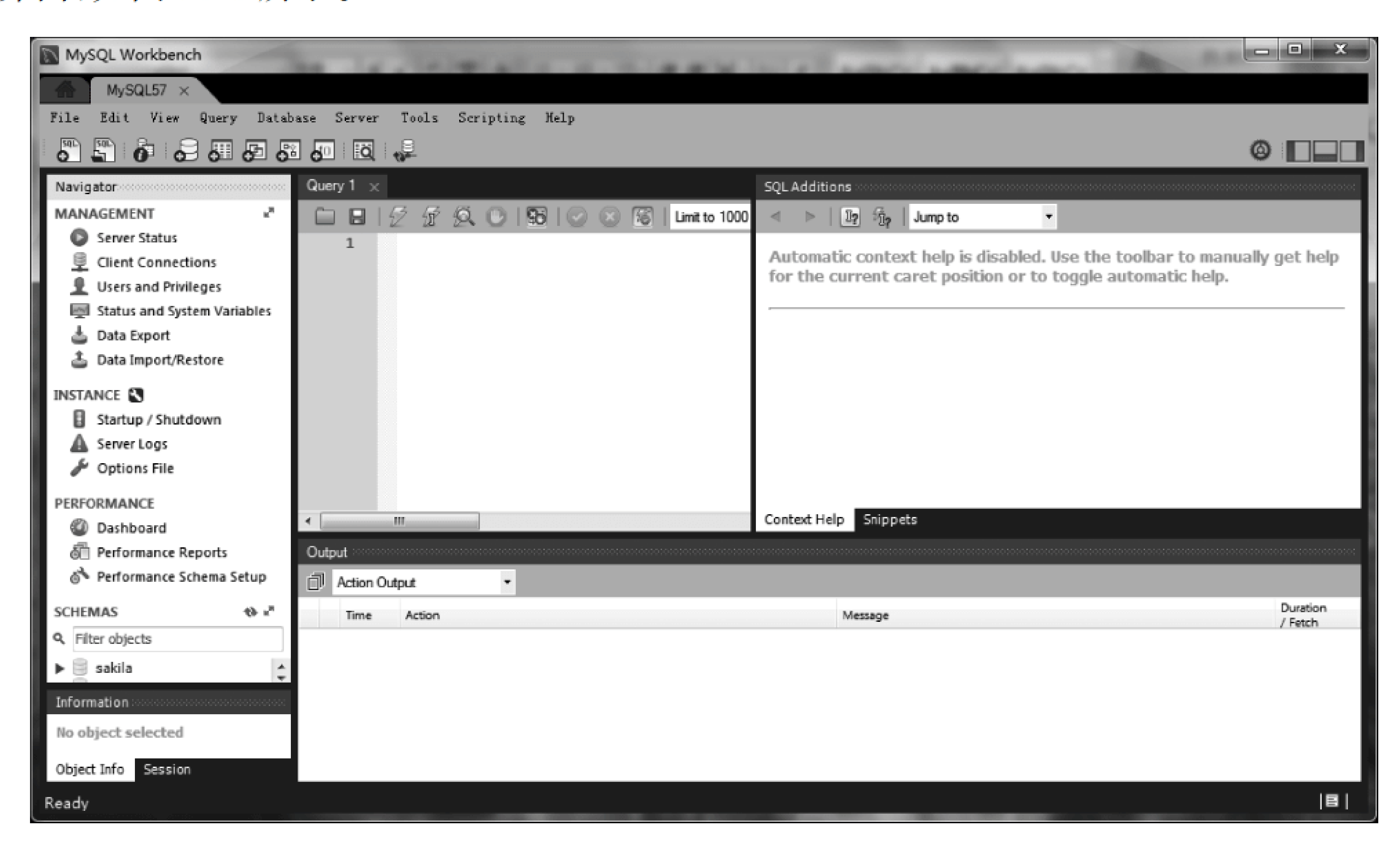


图 18-3　数据库操作主界面

② 创建数据库 teaching。

③ 设置数据库 teaching 的字符集和排序规则参数，也可以使用默认值。

④ 查看数据库。在数据库操作界面的 schemas 区域，就可以看到刚才创建的数据库 teaching 是否存在。

⑤ 删除数据库 info1。

⑥ 设置数据库 teaching 为当前数据库，如图 18-4 所示。

4. 实验报告总结

(1) 创建数据库的默认路径和字符集设置。

(2) 常用系统数据库的功能。

(3) 数据引擎的作用。

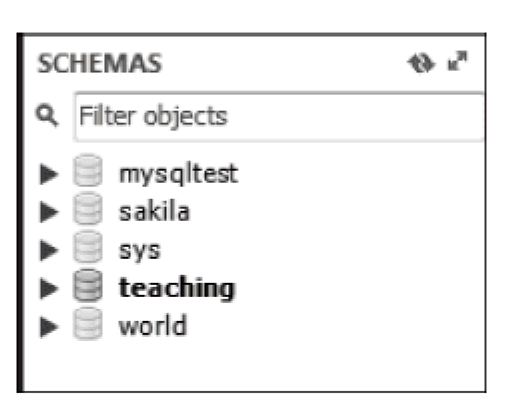


图 18-4　查看数据库

实验 3　数据表的基本操作

1. 实验目的

(1) 掌握创建和修改表结构的方法。

(2) 掌握插入表数据的方法。

(3) 掌握更新表数据的方法。

(4) 掌握删除表数据的方法。

2. 实验预习与准备

预习主教材第 2.2 节和第 4.1～4.3 节：

(1) MySQL 的系统数据类型。

(2) 表结构的创建、修改和删除。

(3) 表数据的浏览、插入、更新和删除。

3. 实验内容及步骤

(1) 创建表结构。如表 18-1 所示，这是教材中的一个 student 表的结构的字段列表。根据该表的内容，学会利用 SQL 语句创建表结构。

表 18-1　student 表结构

列序号	字段名	类型	取值说明	列含义
1	studentno	Char(11)	主键	学生学号
2	sname	Char(8)	否	学生姓名
3	sex	enum (2)	否	性别
4	birthdate	Date	否	出生日期
5	entrance	int(3)	否	入学成绩
6	phone	varchar(12)	否	电话
7	Email	varchar(20)	否	电子信箱

① 登录 MySQL 后，打开数据库 teaching，输入下列代码即可创建 student 表结构：

```
mysql > create table if not exists student
        (
        studentno   char(11) not null,
        sname char(8) not null,
```

```
        sex enum('男', '女') default '男',
        birthdate date not null,
        entrance int(3)  null,
        phone varchar(12) not null,
        Email varchar(20) not null ,
        primary key (studentno)
        );
```

② 查看表和表结构：

```
mysql> show tables;
mysql> describe  student;
```

③ 创建 course 表结构。程序代码如下：

```
mysql> create table if not exists course
        (
        courseno  char(6) not null,
        cname  char(6) not null,
        type char(8) not null,
        period int(2) not null,
        exp int(2) not null,
        term int(2) not null,
        primary key (courseno)
        );
```

④ 创建 score 表结构。程序代码如下：

```
mysql> create table if not exists score
        (studentno  char(11) not null,
        courseno  char(6) not null,
        daily float(3,1) default 0,
        final float(3,1) default 0,
        primary key (studentno , courseno)
        );
```

⑤ 创建 teacher 表结构。程序代码如下：

```
mysql> create table if not exists teacher
        (teacherno  char(6) not null comment '教师编号',
        tname  char(8) not null comment'教师姓名',
        major  char(10) not null comment '专业',
        prof char(10) not null comment '职称',
        department char(16) not null comment '部门',
        primary key (teacherno)
        );
```

⑥ 创建 teach_course 表结构。程序代码如下：

```
mysql> create table if not exists teach_course
        (teacherno char(6) not null,
         courseno  char(6) not null,
         primary key (teacherno,courseno)
        );
```

⑦ 创建选课表 sc。程序代码如下：

```
mysql > create table sc
        (sc_no int(6) not null auto_increment,
         studentno  char(11) not null,
         courseno  char(6) not null,
         teacherno char(6) not null,
         sc_time timestamp not null default now(),
         primary key (sc_no)
        );
```

创建完毕，可以通过安装路径（如：F:\Documents and Settings\All Users\MySQL\MySQL Server 5.7\Data\teaching）查看磁盘文件数据库及其包含的数据表文件。

（2）在可视化工具 MySQL Workbench 中修改表结构。在可视化工具 MySQL Workbench 中展开数据库 teaching 下的表，右击其中一个表，如 course 表，如图 18-5 所示，可以通过执行菜单命令 Alter Table 实现对表结构的修改。其他表的表结构也可以通过这种方式修改，也可以参考教材的相关内容，利用命令方式修改表结构。

① 选择 course 表执行菜单命令 Alter Table，如图 18-6 所示。

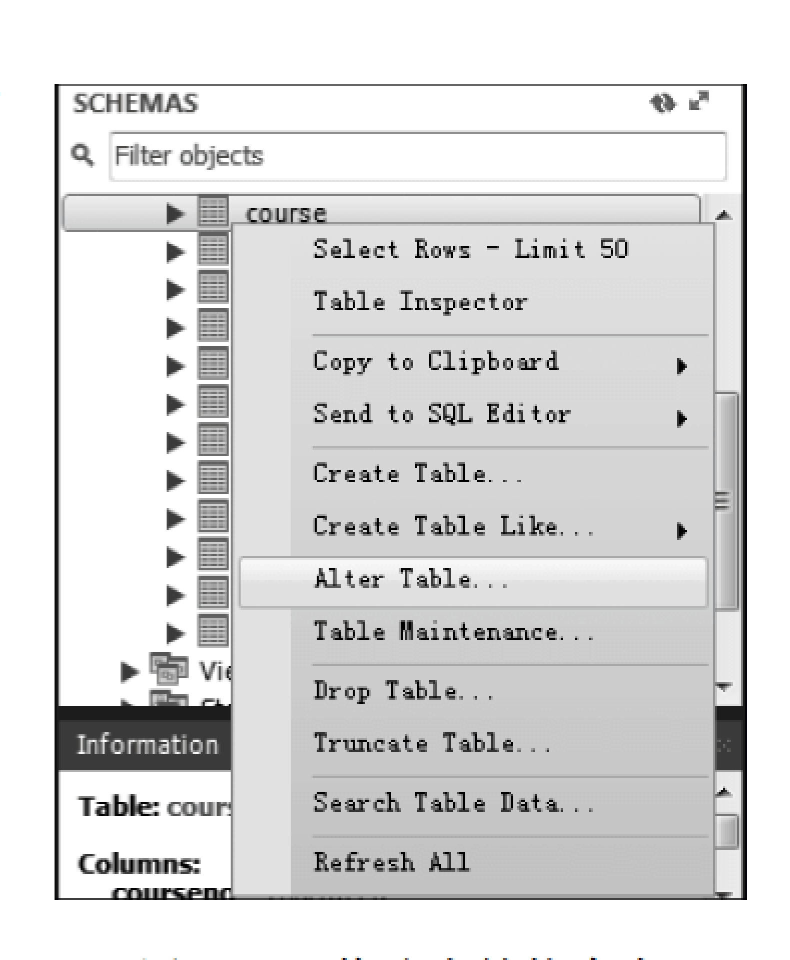


图 18-5　修改表结构命令

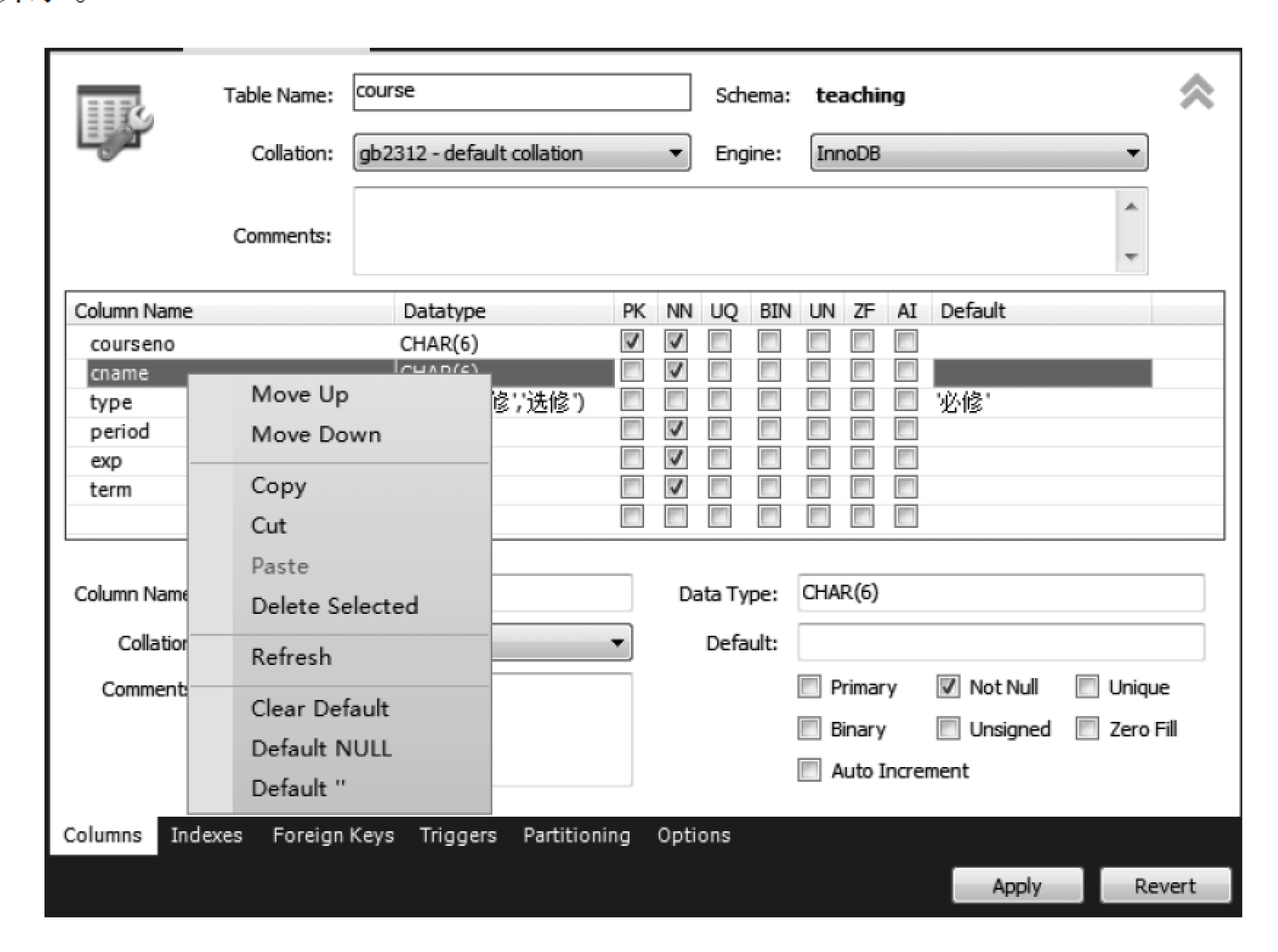


图 18-6　修改表结构

② 在此可以修改表名、默认字符集序列和数据库引擎等表属性。

③ 可以修改字段名、字段类型、字段长度等字段属性。

④ 可以利用弹出菜单移动字段顺序，复制、剪切和粘贴字段，也可以删除、增加和修改字段，清除默认值等。

⑤ 还可以修改约束条件、默认值和字段的自动增长等字段属性。

⑥ 修改完毕，单击 Apply 按钮，就可以保存表结构的修改内容。

(3) 录入表数据。数据库 teaching 的常用表的比较完整的数据可以参考本书第 4 章的数据表数据附录的内容。也可以参考教材的例 4.12～例 4.18 的内容。

① 利用 insert 向表 student 中插入 1 行数据：

```
mysql> insert into student
    ->(studentno,sname,sex,birthdate,entrance,phone,Email)
    -> values ('18122210009','许东山','男','1999/11/5',789,
    -> '13623456778','qwe@163.com');
```

② 利用 insert into 命令向表 student 中插入多行数据：

```
mysql> insert into student values
    -> ('18122221324','何白露',
    -> '女','2000/12/4','879','13178978999','heyy@sina.com '),
    -> ('18125111109','敬横江',
    -> '男','2000/3/1','789','  15678945623','jing@sina.com '),
    -> ('18125121107','梁一苇',
    -> '女','1999/9/3','777','  13145678921','bing@126.com ');
```

③ 利用 replace into 命令向表 course 中插入多行数据：

```
mysql> replace into course values
    -> ('c05103','电子技术','必修','64','16','2'),
    -> ('c05109','C 语言','必修','48','16','2'),
    -> ('c05127','数据结构','必修','64','16','2');
```

④ 可以利用 load data 语句将数据装入数据库表中，实际上这是将其他类型数据导入到数据表的操作。假设 teacher 表的数据已放在“d:\teacher.txt”中，就可以执行如下命令导入数据。注意导入文件路径采用“\\”方式，即转义字符方式。

```
mysql> load data local infile "d:\\teacher.txt" into table teacher;
```

⑤ 利用 set 子句向 score 表插入数据：

```
mysql> insert into score
    -> set studentno = '19120000111',
    -> courseno = 'c05109',daily = 90,final = 95;
```

⑥ 利用现有的表创建和复制成新表，新表中含有全部或部分字段和数据：

```
mysql> create  table  stud  as select *  from student;
```

⑦ 利用可视化工具 MySQL Workbench 添加数据。选择要添加数据的表 student，在图 18-5 所示的菜单中，执行 Select Rows 命令，如图 18-7 所示，即可在表中添加数据。添加完毕，单击 Apply 按钮，保存即可。如果放弃添加，单击 Revert 按钮即可。



图 18-7　添加表数据

（4）修改表数据。

① 修改指定记录的数据：

```
mysql > update score set daily = 95
    -> where studentno = '18137221508' && courseno = 'c08126';
```

② 成批修改记录：

```
mysql > update stud set entrance = entrance * 1.08 where entrance < 600;
```

③ 利用可视化工具 MySQL Workbench 修改数据。选择要修改数据的表 teacher，执行 Select Rows 命令，如图 18-8 所示，即可在表中修改数据，如修改字段值“计算机应用”为“计算机技术”。修改完毕，关闭窗口，在新对话框中单击 Apply 按钮，保存即可。如果放弃添加，单击 Revert 按钮即可。

（5）删除表数据。

① 删除符合条件的指定记录行。删除 student 表中入学成绩低于 600 分的记录：

```
mysql > delete  from student where entrance < 600;
```

② 利用 limit 子句删除指定记录行。删除 student 表中入学成绩最低的 2 行记录：

```
mysql > delete  from student order by entrance limit 2;
```

③ 利用可视化工具 MySQL Workbench 删除数据。在如图 18-9 所示的菜单中，执行 Delete Rows 命令，即可删除指定记录行。

④ 清空表记录，即一次性删除表中的所有记录。如图 18-10 所示，执行 Truncate Table 命令，即可删除选定的所有记录。

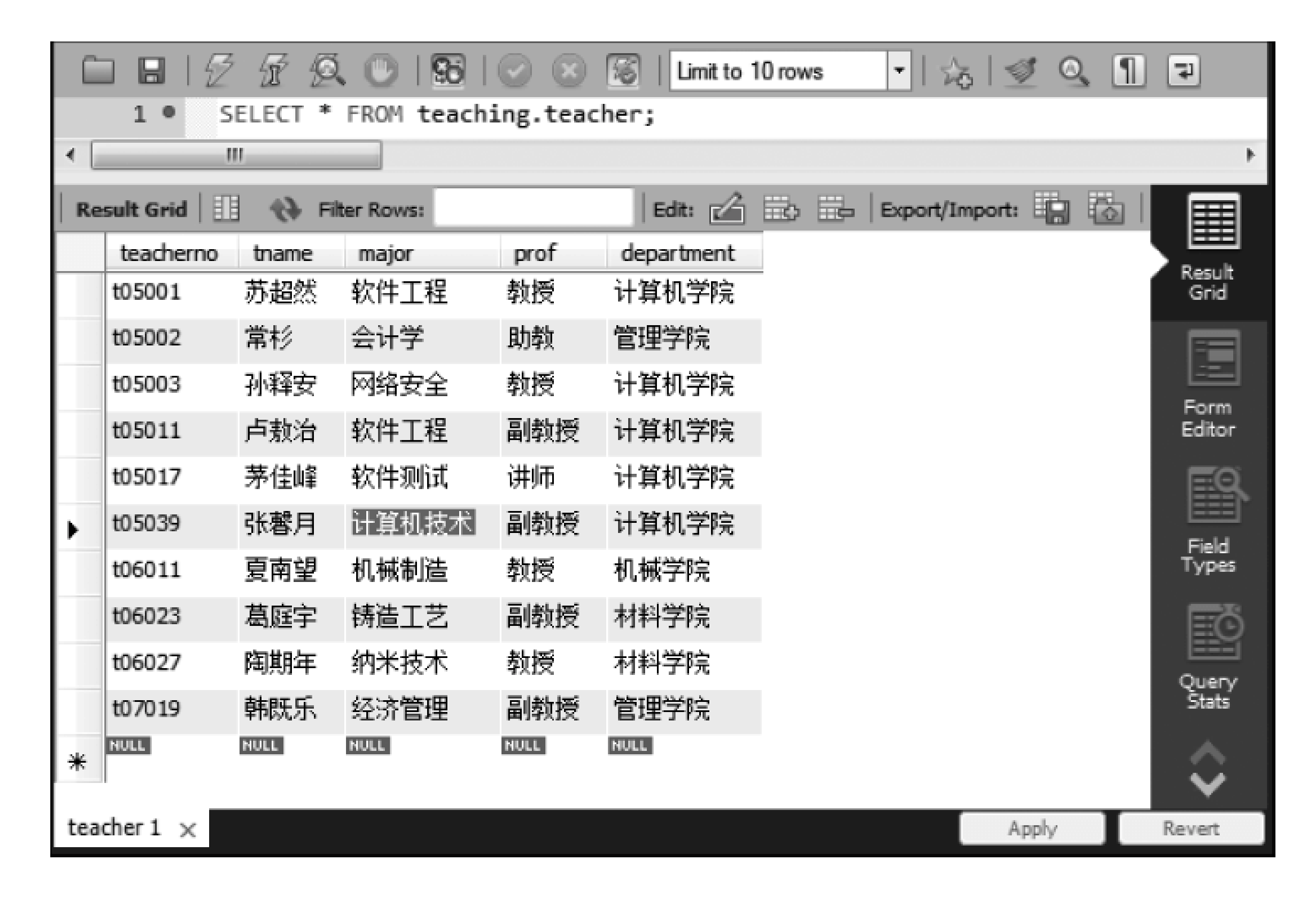


图 18-8　修改表数据

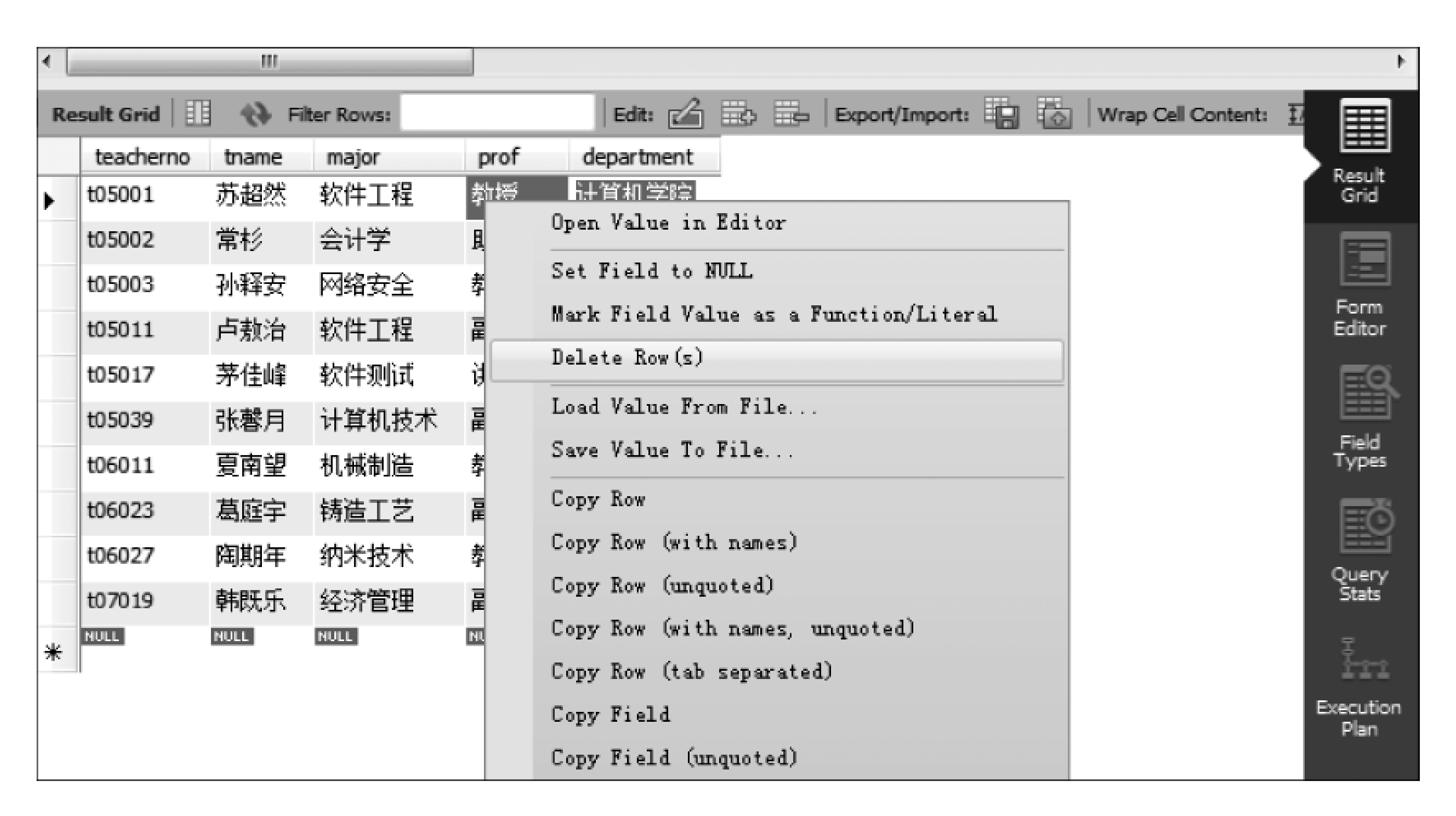


图 18-9　删除表数据

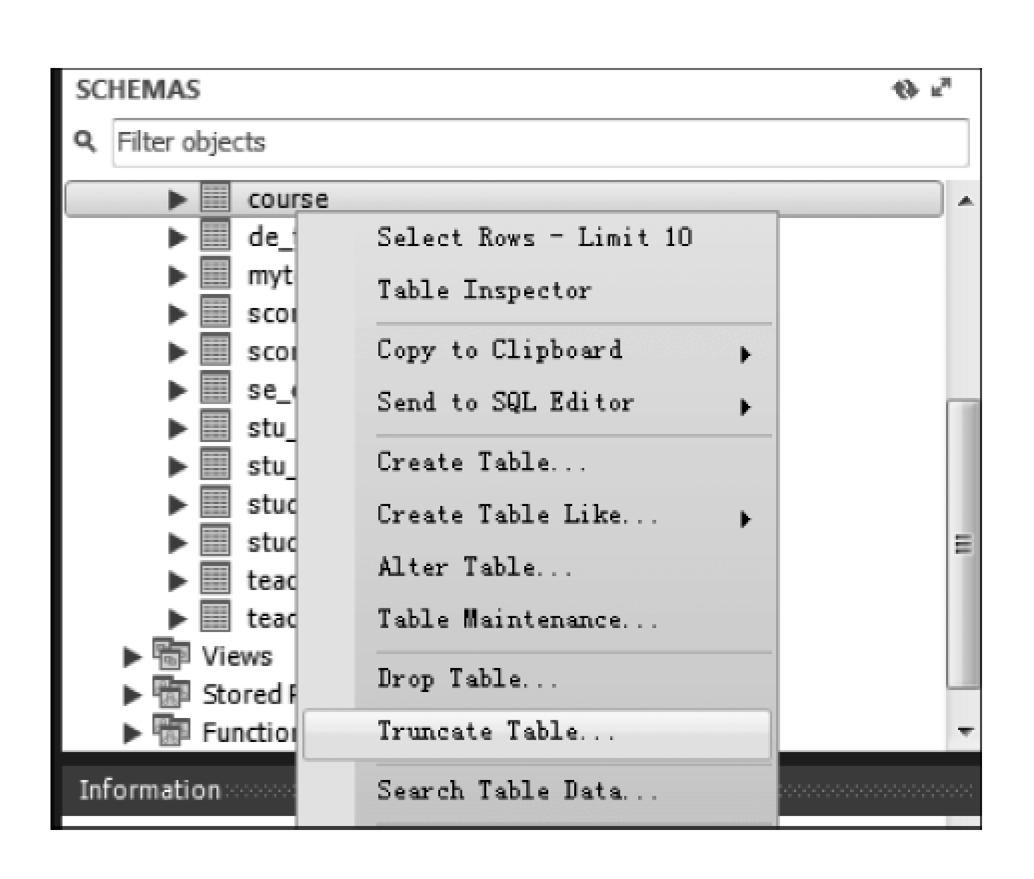


图 18-10　清空表数据

当然,在 DOS 窗口中执行如下命令也可以删除指定表 stud 的所有数据。

```
mysql > truncate table stud;
```

4. 实验报告总结

(1) 创建表结构的方法总结。

(2) 修改表结构对表中记录的影响。

(3) 利用可视化工具 MySQL Workbench 管理表的优势。

实验 4　数据完整性的基本操作

1. 实验目的

(1) 理解实现数据完整性的概念。

(2) 掌握创建、修改主键约束的方法。

(3) 掌握利用 MySQL 语句创建其他约束的方法。

2. 实验预习与准备

预习主教材第 4.4 节:

(1) 数据完整性的类型和实现机制。

(2) 5 类约束的创建和修改。

3. 实验内容及步骤

(1) 主键约束的创建与应用。

① 利用 course 表创建一个新表 course1:

```
mysql > create  table  course1 as select *  from course;
```

② 创建 course01 的主键约束,字段名为 cname:

```
mysql > alter table course1 add primary key (cname);
```

③ 删除主键约束,重新创建主键约束,字段名为 courseno:

```
mysql > alter table  course1 drop primary key;
mysql > alter table  course1  add primary key (courseno);
```

④ 验证主键约束,观察主键值重复时的数据插入情况:

```
mysql > insert into course1 values
    -> ('c05188','电子技术','必修','64','16','2');
mysql > insert into course1 values
    -> ('c05188','控制原理','必修','64','20','3');
```

(2) 外键约束的创建和应用。

① 利用 score 表创建一个新表 score1:

```
mysql > create  table  score1 as select *  from score;
```

② 对表 score01 添加外键:

```
mysql > alter table score1
    -> add constraint fk_cou_score
    -> foreign key(courseno) references course1(courseno);
```

③ 验证外键约束,观察外键值对数据插入情况的影响:

```
mysql> insert into score1 values
    -> ('18137221508','c05109',94,92);
mysql> insert into score1 values
    -> ('18137221508','c11111', 99,98);     //course1 表中无'c11111'课程号
```

(3) 检查约束的创建和应用。

① 创建检查约束,final 值以 0~100 为界值:

```
mysql> alter table score1
     -> add constraint sc_check1 check(final >= 0 and final <= 100);
```

② 验证检查约束,观察合法与非法数据插入的情况:

```
mysql> insert into score1 values
    -> ('19126113307','c05109',94,100);
mysql> insert into score1 values
    -> ('19122203567','c05108',94,101);   //final 值超界
```

(4) 唯一约束的创建和应用。

① 创建唯一约束:

```
mysql> alter table student add unique(phone);
```

② 验证唯一约束,观察重复电话数据插入的情况:

```
mysql> select  studentno, sname, phone from student;
mysql> insert into student values
    -> ('19126113306','梅青','女','2001/9/7','787','
    -> 13245678543','mei@163.com');
```

(5) 非空约束的应用。主键字段值不允许为 null,输入空值观察结果:

```
mysql> insert into course1 values
    -> (null,'网络技术','必修','64','16','2');
```

(6) 利用可视化工具 MySQL Workbench 管理表的约束。在 MySQL Workbench 窗口中修改 score1,如图 18-11 所示。可以通过表结构中字段后面的复选按钮管理主键约束、唯一约束和非空约束。如果单击图形下方的 foreign keys 标签,还可以管理外键约束。

4. 实验报告总结

(1) 不同类型数据完整性的特点和实现方法。

(2) 各种约束的功能及对表数据修改的影响。

实验 5　MySQL 数据的查询基本操作

1. 实验目的

(1) 掌握 select 语句的格式和各子句的功能。

(2) 掌握 where 子句中 like、in、between...and、is 等逻辑运算符的使用。

(3) 掌握 group by 语句和聚合函数的使用。

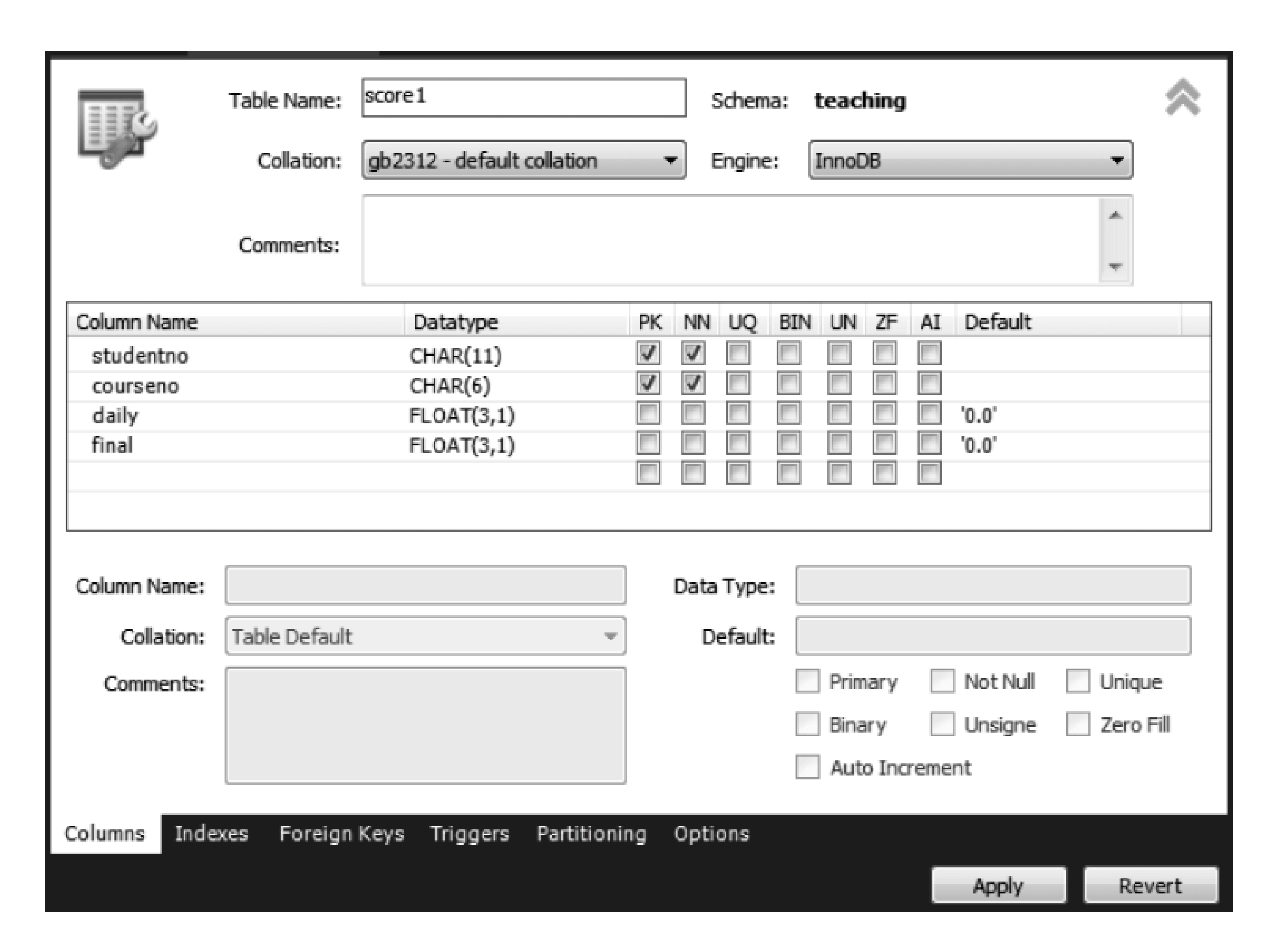


图 18-11　完整性约束

2. 实验预习与准备

预习主教材第 5.1～5.3 节：

(1) 利用 select...from 子句实现投影查询和设置结果集格式。

(2) 利用 where 子句实现数据过滤。

(3) 利用 group by 子句和 having 子句实现分组查询。

(4) group by 语句和聚合函数的使用。

(5) 利用 limit 子句输出部分记录。

3. 实验内容及步骤

(1) select...from...where 基本子句的使用。

① 在 student 表中查询出生日期在 2003 年以前的学生的学号、姓名、电话和年龄：

```
mysql> select  studentno 学号,sname 姓名,
    -> phone 电话,year(now()) - year(birthdate) 年龄
    -> from  student
    -> where  birthdate<'2003-01-01';
```

② 在 score 表中查询期中成绩高于 90 分的学生的学号和课程号,并按照学号排序：

```
mysql> select  distinct studentno,courseno
    -> from   score
    -> where daily>90
    -> order  by studentno;
```

③ 查询学号分别为 18122221324、18137221508 和 18137221508 学生的课程号、平时成绩和期末成绩：

```
mysql> select studentno,courseno ,daily ,final
    -> from score
    -> where studentno in('18122221324','18137221508','18137221508');
```

④ 查询选修课程号为 c08123 的学生的学号和期末成绩，并且要求平时成绩在 85～100 分。

```
mysql > select studentno, final
    -> from score
    -> where courseno = 'c08123' and daily between 85 and 100;
```

⑤ 在 student 表中显示所有姓赵的学生的姓名、生日和 Email：

```
mysql > select  sname, birthdate, Email
    -> from  student
    -> where  sname  like '赵 % ';
```

⑥ 在 score 表中显示期中成绩高于 85 分、期末成绩高于 90 分的学生的学号、课程号和成绩：

```
mysql > select studentno,courseno,daily,final
    -> from score
    -> where daily >= 85 and final >=  90;
```

⑦ 查询计算机学院专业为“软件工程”或“网络技术”的教师的教师号、姓名和职称：

```
mysql > select teacherno,tname, prof
    -> from teacher
    -> where department = '计算机学院'
    -> and (major  = '软件工程'or major  = '网络技术');
```

(2) 排序、分组和限定记录的查询。

① 在 student 表中输出高于 800 分的学生的学号、姓名、电话和入学成绩，并按照入学成绩的降序排列：

```
mysql > select studentno ,sname ,phone,entrance
    -> from student
    -> where entrance > 800
    -> order by entrance  desc;
```

② 在 score 表中查询总评成绩大于 85 分的学生的学号、课程号、电话和总评成绩，并先按照课程号的升序、再按照总评成绩的降序排列。总评成绩计算公式如下：

总评成绩＝daily＊0.2＋final＊0.8

```
mysql > select courseno,daily  * 0.2 +  final * 0.8 as '总评',phone,studentno
    -> from score
    -> where daily  * 0.2 +  final * 0.8 > 85
    -> order by courseno, daily  * 0.2 +  final * 0.8 desc;
```

③ 利用 group by 子句对 score 表数据分组，显示每个学生的学号和平均总评成绩。总评成绩计算公式如下：

总评成绩＝daily＊0.3＋final＊0.7

```
mysql > select studentno , round(avg(daily * 0.3 + final * 0.7),2) as avg1
    -> from score
    -> group by studentno;
```

④ 使用 group by 关键字和 group_concat()函数对 score 表中的 courseno 字段进行分组查询。可以查看该学生选修的课程号：

```
mysql> select studentno 学号,group_concat(courseno) 选课学生
    -> from score
    -> group by studentno ;
```

⑤ 查询选课在 2 门以上且各门课程期末成绩均高于 85 分的学生的学号及其总成绩，查询结果按总成绩降序列出：

```
mysql> select studentno 学号,sum(daily * 0.3 + final * 0.7) as '总分'
    -> from  score
    -> where final >= 85
    -> group by  studentno
    -> having count( * )>= 2
    -> order by sum(daily * 0.3 + final * 0.7) desc;
```

⑥ 查询 score 表中期末成绩 final 高于 90 分的成绩，按照平时成绩 daily 进行升序排列，从编号 1 开始，查询 3 条记录：

```
mysql> select  *  from  score
    -> where  final > 90
    -> order  by  daily  asc
    -> limit  1,3;
```

(3) 聚合函数的应用。

① 查询 score 表中学生的期末总成绩大于 275 分的学生学号、总成绩及平均成绩：

```
mysql> select studentno 学号, sum(final) 总分, avg(final)平均分
    -> from score
    -> group by studentno
    -> having sum(final)> 275
    -> order by studentno;
```

② 查询选修课程号为 c05103 的学生的期末最高分、最低分及之间相差的分数：

```
mysql> select  max(final) 最高分, min(final) 最低分,
    -> max(final) - min(final) as  分差
    -> from score
    -> where (courseno = 'c05103');
```

③ 查询 score 表中每个学生的期末平均值和所有成绩的平均值：

```
mysql> select studentno 学号,avg(final) 平均分
    -> from score
    -> group by studentno with  rollup;
```

4. 实验报告总结

(1) 逻辑运算符在 where 子句中的使用方法。

(2) 利用带有通配符的 like 实现模糊查询的特点。

(3) group by 子句在数据统计中的应用。

实验 6　多表连接和子查询

1. 实验目的

(1) 掌握多表连接查询、子查询的基本概念。

(2) 掌握多表连接的各种方法，包括内连接、外连接、交叉连接等。

(3) 掌握子查询的方法，包括相关子查询和不相关子查询。

(4) 掌握正则表达式在查询中的作用。

2. 实验预习与准备

预习主教材第 5.5 和第 5.6 节：

(1) 利用多表连接方式查询数据的 3 种基本格式。

(2) 利用子查询方式查询数据。

(3) 利用正则表达式进行模糊查询的基本过程。

3. 实验内容及步骤

(1) 查询选修课程号为 c06108 的学生的学号、姓名和期末成绩：

```
mysql> select student.studentno,sname,final
    -> from   student inner join score
    -> on student.studentno = score.studentno
    -> where  score.courseno = 'c06108';
```

(2) 利用左外连接方式查询学生的学号、姓名、平时成绩和期末成绩：

```
mysql> select  student.studentno,sname,daily,final
    -> from  student  left join score
    -> on student.studentno = score.studentno;
```

(3) 利用右外连接方式查询教师的排课情况：

```
mysql> select teacher.teacherno,tname, major, courseno
    -> from  teacher right join teach_course
    -> on teacher.teacherno = teach_course.teacherno;
```

(4) 查询 19 级学生的学号、姓名、课程名、期末成绩及学分：

```
mysql> select student.studentno,sname,cname,final,round(period/16,1)
    -> from score  join student on  student.studentno = score.studentno
    ->             join course on  score.courseno = course.courseno
    -> where  left(student.studentno,2) = '19';
```

(5) 查询期末成绩高于 90 分、总评成绩高于 85 分的学生的学号、课程号和总评成绩：

```
mysql> select TT.studentno 学号 ,TT.courseno 课程号 ,
    ->        TT.final * 0.8 + TT.daily * 0.2  总评
    -> from  (select *  from score  where final > 90) as TT
    -> where TT.final * 0.8 + TT.daily * 0.2 > 85;
```

(6) 查询期末成绩比选修该课程平均期末成绩低的学生的学号、课程号和期末成绩：

```
mysql> select studentno,courseno,final
```

```
    -> from score as a
    -> where final < (select avg(final)
    ->                 from score as b
    ->                 where a.courseno = b.courseno
    ->                 group by  courseno );
```

(7) 获取期末成绩中含有高于 90 分的学生的学号、姓名、电话和 E-mail：

```
mysql> select studentno,sname,phone,Email
    -> from student
    -> where studentno in ( select studentno
    ->                      from score
    ->                      where final>90);
```

(8) 查找 score 表中所有比 c05103 课程期末成绩都高的学号、姓名、电话和期末成绩：

```
mysql> select student.studentno,sname, phone,final
    -> from score inner join student
    ->      on score.studentno =  student.studentno
    -> where  final>all
    ->         (select final from score where courseno =  'c05103');
```

(9) 将 student 表中 1999 年以后出生的学生记录添加到 stud 表中：

```
mysql> insert into  stud
    ->      ( select  * from student
    ->        where birthdate>= '1999-12-31');
```

(10) 查询 student 表中学生电话号码尾数为 8 的学生的部分信息：

```
mysql> select  studentno, sname, phone, Email
    -> from student
    -> where phone regexp '8$';
```

4. 实验报告总结

(1) 内连接与外连接的功能及区别。

(2) 相关子查询与不相关子查询的功能及区别。

(3) 使用正则表达式进行模糊查询的基本过程和方法。

实验 7　索引和视图

1. 实验目的

(1) 掌握索引和视图的基本概念和功能。

(2) 掌握利用 MySQL 语句创建、维护索引的方法。

(3) 掌握利用 MySQL 语句创建、修改视图的方法。

(4) 掌握通过视图插入、修改、删除基本表中数据的方法。

2. 实验预习与准备

预习主教材第 6 章：

(1) 利用 MySQL 语句创建、维护索引。

(2) 利用 MySQL 语句创建、修改视图。

(3) 通过视图查看和修改基本表中的数据。

3. 实验内容及步骤

(1) 索引的创建与管理。

① 为 student 表的 phone 列上建立一个降序普通索引 phone_idx：

```
mysql> create index phone_idx  on student(phone desc);
```

② 在 course 表的 cname 列上建立一个唯一性索引 cname_idx：

```
mysql> create  unique  index  cname_idx on course (cname);
```

③ 在 score 表的 studentno 和 courseno 列上建立一个复合索引 stu_cour_idx：

```
mysql> create  index  stu_cour_idx  on  score(studentno,courseno);
```

④ 在 teacher 表上建立 teacherno 主键索引，建立 tname 和 prof 的复合索引：

```
mysql> alter table teacher
    -> add primary key(teacherno),
    -> add index mark(tname, prof);
```

⑤ 删除 teacher 表的 mark 索引：

```
mysql> drop index mark on teacher1;
```

⑥ 利用 alter table 语句同样可以删除 cname_idx 索引：

```
mysql> alter table  course  drop  index  cname_idx;
```

(2) 利用 MySQL Workbench 创建索引。

① 进入修改表 student 界面。

② 打开 index 选项卡，在 Index Name 的文本框中输入索引名称 un_phone，右侧的 Index Columns 会自动显示表 student 中的所有列名，选择 phone 列。

③ 存储类型选择 BTREE，选择索引类型 unique，表示创建唯一性索引，其他参数采用默认值，如图 18-12 所示。

④ 设置完成后，单击 Apply 按钮，出现脚本对话框。

⑤ 单击 Apply 按钮，进入完成对话框，单击 Finish 按钮，即可完成在数据库 teaching 中 student 表上的唯一索引 un_phone 的创建。

⑥ 也可以在此界面中实现索引的修改和删除操作。

(3) 创建和管理视图。

① 在 teacher 表上创建一个简单的视图，视图名称为 v_teacher：

```
mysql> create  view v_teacher
    -> as select *  from  teacher;
mysql> select *  from  v_teacher;
```

② 在 student 表、course 表和 score 表上创建一个名为 stu_score 的视图。视图中保留 19 级的女生的学号、姓名、电话、课程名和期末成绩：

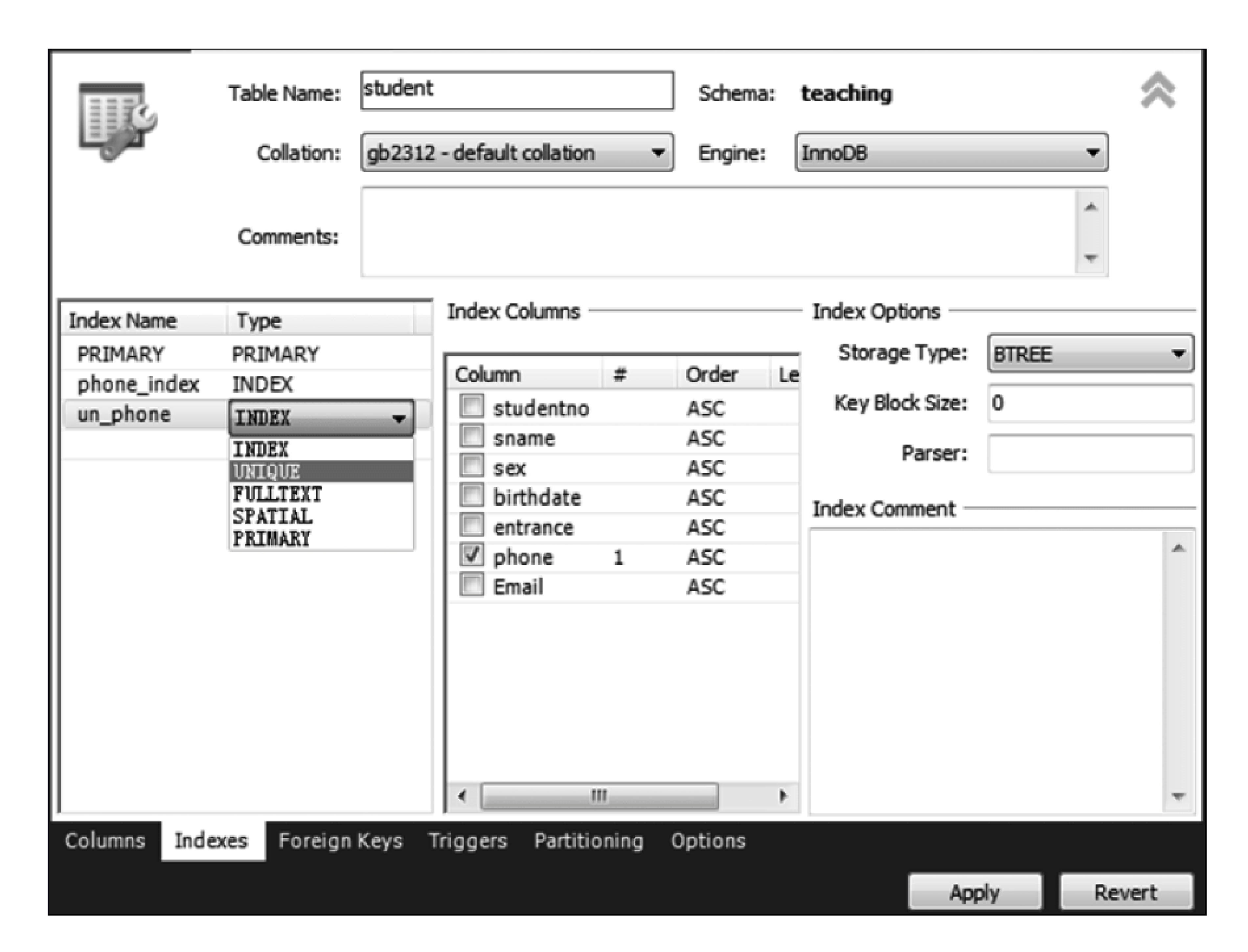


图 18-12　创建唯一索引

```
mysql> create  view  stu_score
    -> as select  student.studentno, sname, phone, cname,final
    -> from  score  join student  on  student.studentno = score. studentno
    ->              join  course  on  course.courseno = score.courseno
    -> where  sex = '女'  and  left(student.studentno,2) =  '19';
mysql> select * from  stu_score;
```

③ 创建视图 v_teach，统计材料学院的教师中不是教授或副教授的教师号、教师名和专业：

```
mysql> create view  v_teach
    -> as select  teacherno, tname, major
    -> from  teach_view1
    -> where  prof not like  '%教授'  and  department = '材料学院';
```

④ 查看视图定义情况：

```
mysql> show create view stu_score;
```

⑤ 修改视图 v_teach，统计材料学院的教师中的教授和副教授的教师号、教师名和专业，并在视图名后面指明视图列名称：

```
mysql> alter view  v_teach (教师号,教师名,专业)
    -> as select  teacherno, tname, major
    -> from  teach_view1
    -> where  prof  like  '%教授'  and  department = '材料学院';
mysql> select  *  from  v_teach;
```

⑥ 删除视图 v_teach 的命令：

```
mysql > drop view v_teach;
```

(4) 利用 MySQL Workbench 管理视图。

① 启动 MySQL Workbench 工具，选择 teaching 数据库，展开 Views 选项，在如图 18-13 所示的弹出菜单中，执行 Create View 命令，进入文本编辑区。

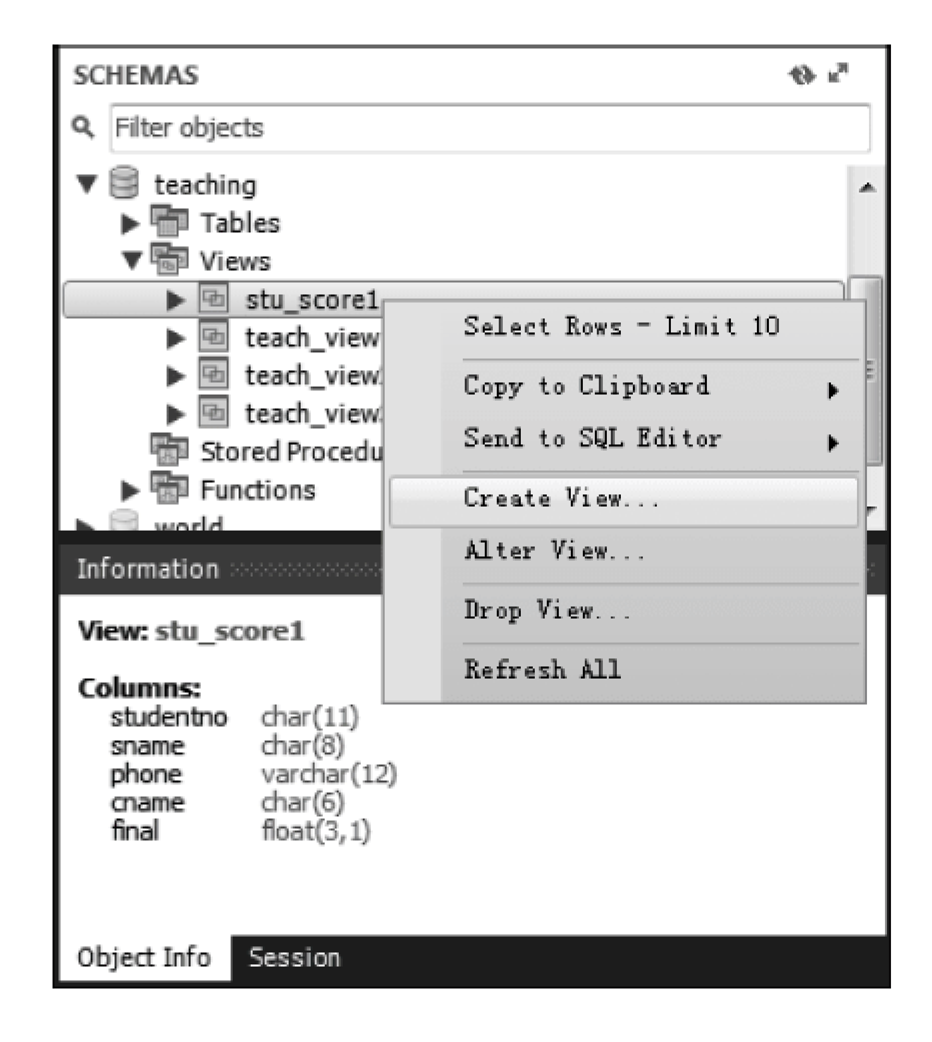


图 18-13　创建视图

② 在文本编辑区，输入创建视图 st_view 的代码。自己检查无误后，单击 Apply 按钮。

③ 在弹出的代码对话框中，单击 Apply 按钮，单击 Finish 按钮完成视图创建过程。

④ 展开 Views 文件夹，选择视图 st_view，执行 Select　Rows-Limit 10 命令，即可观察到输出结果。

⑤ 在如图 18-13 所示的弹出菜单中，还可以通过执行 Alter View 命令、Drop View 命令来修改和删除视图。

(5) 利用视图修改表数据。

① 创建视图 view_avg，统计各门课程的平均成绩，并按课程名称升序排列：

```
mysql > create  view  view_avg
    -> as select  cname  课程名,  avg(final)  平均成绩
    -> from  score join course  on  score.courseno = course.courseno
    -> group  by  cname  asc;
mysql > select * from view_avg;
```

② 通过视图 v_teacher，对基表 teacher 进行插入、更新和删除数据的操作：

```
mysql > insert into v_teacher (teacherno, tname, major, prof, department)
    -> values ('t07027', '陶颐年', '合金技术', '副教授', '材料学院');
mysql > update v_teacher  set  prof  =  '副教授'  where  teacherno  =  't07019';
mysql > delete  from  v_teacher  where  teacherno  =  't07027';
```

③ 视图 stu_score 依赖于源表 student、course 和 score 3 张表，包括 studentno、sname、phone、cname 和 final 5 个字段，通过 stu_score 修改基本表 student 中的学号为 19112100072

的电话号码：

```
mysql > update stu_score set phone = '133123456777'
    -> where   studentno   = '19112100072';
mysql > select   studentno, sname, phone   from   student
    -> where   studentno   = '19112100072' ;
# stu_score 是针对 19 级的视图,验证能否修改 18 级数据
mysql > update stu_score set phone = '135123456777'
    -> where   studentno   = '18125121107';
```

④ 修改 v_teacher 的视图定义，添加 with check option 选项：

```
mysql > alter view v_teacher
    -> as
    -> select *    from teacher
    -> where department  =  '计算机学院'
    -> with check option ;
```

4. 实验报告总结

(1) 不同类型索引对表数据进行查询的功能。

(2) 视图与数据表及查询的区别。

(3) 通过视图插入、修改、删除基本表中数据的注意事项。

实验 8　MySQL 函数的创建和使用

1. 实验目的

(1) 掌握自定义函数的格式、功能和调用过程。

(2) 掌握 MySQL 控制流语句的基本功能和分类。

(3) 掌握利用控制流语句实现基本的分支选择和循环处理的功能。

(4) 了解其他控制流语句的功能和应用。

2. 实验预习与准备

预习主教材第 7 章。

(1) 利用 begin...end 语句定义语句块的场合。

(2) 利用 if...else 语句和 case 语句实现分支选择的使用方法。

(3) 利用循环语句实现复杂算法处理的过程。

3. 实验内容及步骤

(1) 会话变量等基本操作的使用。

① 使用查询结果给变量赋值：

```
mysql > set @sname = (select sname from student
    -> where studentno = '18135222201');
mysql > select studentno, sname, birthdate
    -> from student where sname = @sname;
```

② 利用 select 语句将表中数据赋值给变量：

```
mysql > select @sname := sname from student limit 5,1;
```

③ 利用 select 语句输出表达式：

```
mysql> select 1 + 7.5 + 3.4, '2018 - 01 - 20' +  interval 6 month;
```

④ 改变 MySQL 命令的结束标记：

```
mysql> delimiter  $$
mysql> select studentno,sname,birthdate
    -> from student where sname like '梅 % ' $$
mysql>  delimiter ;
```

⑤ 利用预处理 SQL 语句输出 course 表中的前 5 行记录的部分数据：

```
mysql> set @a = 5;
mysql> prepare  STMT
    -> from "select courseno,cname,type, period  from  course  limit ?";
mysql> execute STMT using @a;
```

(2) 自定义函数的创建和调用。

① 创建一个名为 f_course 的函数返回表 course 中的指定课程号的课程名：

```
mysql> delimiter &&
mysql> create  function f_course(c_no varchar(6))
    -> returns  char(6)
    -> begin
    ->  return  (select  cname  from  course
    ->              where  courseno  = c_no);
    -> end &&
mysql> delimiter ;
```

② 调用函数 f_course：

```
mysql> select f_course('c05109');
```

③ 查看指定数据库(如 teaching)中的所有自定义函数名：

```
mysql> select name from MySQL.proc
    -> where db  =  'teaching' and type = 'function';
```

④ 查看指定函数 f_course 的信息：

```
mysql> select  *  from information_schema.routines
    -> where routine_name = 'f_course';
```

⑤ 修改存储函数 f_course 的定义。将读写权限改为 reads sql data，并加上注释信息“find function name”：

```
mysql> alter  function  f_course
    -> reads sql data
    -> comment  'find function name';
mysql> select  SPECIFIC_NAME,SQL_DATA_ACCESS,
    -> routine_comment  from  information_schema.Routines
    -> where  routine_name = 'f_course';
```

⑥ 练习利用 MySQL Workbench 工具创建、修改和删除过程函数。

(3) 条件控制语句的应用。

① 删除表 course 中指定课程号在 score 中的记录：

```
mysql> delimiter $$
mysql> create function delete_cno (c_no char(6))
    -> returns char(6)
    -> begin
    -> declare c_name char(6);
    -> select cname into c_name from course where courseno = c_no;
    -> if c_name  is null then
    ->   delete from score where courseno = c_no;
    -> return 'yes';
    -> else
    -> return 'no';
    -> end if;
    -> end $$
mysql> delimiter ;
```

② 创建函数 ex_case，通过 case 语句首先判断传入参数的值是否为 100，如果条件成立则输出 1，如果条件不成立则再判断该传入参数的值是否为 200，如果成立则输出 2，如果条件不成立则再判断该传入参数的值是否为 300，如果成立则输出 3，当以上条件都不满足时输出 0。

```
mysql> delimiter //
mysql> create function ex_case( x int)
    -> returns  int
    -> begin
    -> case  x
    -> when 100  then set  x = 1;
    -> when 200  then set  x = 2;
    -> when 300  then set  x = 3;
    -> else  set  x = 0;
    -> end case;
    -> return  x;
    -> end //
mysql> delimiter ;
mysql> select exam_case(127);
```

(4) 循环语句的应用。

① 定义函数 ex_while，应用 while 语句求 50～100 的和：

```
mysql> delimiter //
mysql> create  function  ex_while(n int) returns int
    -> begin
    -> declare sum int default 0;
    -> declare m int default 50;
    -> while  m < = n do
    -> set  sum = sum + m;
    -> set  m = m + 1;
```

```
    -> end while;
    -> return  sum;
    -> end //
mysql > delimiter ;
mysql > select exam_while(100);
```

② 定义函数 ex_loop,应用 loop 语句求 20～70 的和。通过 leave 语句退出循环并输出结果:

```
mysql > delimiter//
mysql > create  function  exam_loop(n int) returns int
    -> begin
    -> declare  sum int default 0;
    -> declare  m int default 20;
    -> loop_label:loop
    -> set sum = sum + m;
    -> set m = m + 1;
    -> if  m > n  then
    -> leave  loop_label;
    -> end if;
    -> end loop;
    -> return  sum;
    -> end //
mysql > delimiter;
mysql > select exam_loop(70);
```

4. 实验报告总结

(1) if...else 语句和 case 语句实现分支选择的区别。

(2) while、loop、repeat 等循环语句的功能和区别。

(3) 嵌套 if...else 语句的特点和使用注意事项。

实验 9　存储过程的创建和使用

1. 实验目的

(1) 掌握存储过程和触发器的基本概念和功能。

(2) 掌握创建、管理存储过程的方法。

(3) 掌握创建、管理触发器的方法。

2. 实验预习与准备

预习主教材第 8.1 节:

(1) 存储过程的基本概念。

(2) 利用 MySQL 语句创建、管理存储过程。

(3) 利用 MySQLWorkbench 工具创建、管理存储过程。

3. 实验内容及步骤

(1) 创建存储过程 p_student,从 student 表中检索出所有电话以 135 开头的学生的学号、姓名、出生日期和电话等信息:

```
mysql > delimiter //
```

```
mysql> create procedure  p_student()
    -> reads sql data
    -> begin
    -> select studentno, sname, birthdate , phone
    -> from student
    -> where phone like '135 % ' order by  studentno ;
    -> end //
mysql> delimiter ;
mysql> call p_student();
```

(2) 创建存储过程 pro_select(),用指定的学号和课程号为参数查询学生成绩:

```
mysql> delimiter  $$
mysql> create  procedure  pro_select(in s_no char(11),c_no char(6))
    -> begin
    -> select *   from  score
    -> where studentno = s_no and  courseno =  c_no ;
    -> end  $$
mysql> delimiter ;
mysql> call pro_select('19122203567','c06127');
```

(3) 创建一个存储过程 pro_insert(),向 score 表中插入一行记录,然后创建另一存储过程 pro_out(),调用存储过程 pro_insert(),并查询输出 score 表中插入的记录:

```
#先创建第 1 个存储过程 pro_insert()
mysql> create procedure pro_insert()
    -> insert into score  values('19111133071', 'c08171',100,99);
#创建第 2 个存储过程 pro_out(),调用 pro_insert()
mysql> delimiter  $$
mysql> create  procedure  pro_out()
    -> begin
    -> call pro_insert();
    -> select   *   from  score
    -> where studentno = '19111133071';
    -> end  $$
mysql> call do_out();
```

(4) 查看存储过程的定义:

```
mysql> show  procedure  status like 'pro_ % ';
mysql> show  create  procedure  pro_out;
mysql> select * from information_schema.routines
    -> where  routine_name = 'pro_out ' ;
mysql> show create procedure p_student;
```

(5) 修改存储过程 do_insert()的定义。将读写权限改为 modifies sql data,并指明调用者可以执行:

```
mysql> alter  procedure  do_insert
    -> modifies sql data
    -> sql security invoker;
```

(6) 利用 MySQL Workbench 工具管理存储过程。利用 MySQL Workbench 工具管理存储过程主要包括对存储过程的创建、修改、查看、删除和执行操作。

4. 实验报告总结

(1) 存储过程中输出参数与输入参数在定义、调用时的区别。

(2) 触发器 insert、delete 和 update 操作与临时表 inserted 和 deleted 的关系。

(3) 存储过程与触发器的联系与区别。

实验 10 触发器、游标和事件

1. 实验目的

(1) 掌握游标、事件和触发器的基本概念及功能。

(2) 掌握创建、管理触发器的方法。

(3) 掌握创建、管理事件的方法。

(4) 掌握游标处理结果集的基本过程。

2. 实验预习与准备

预习主教材第 8.2～8.4 节：

(1) 触发器的基本概念及触发器与存储过程的关系。

(2) 利用 MySQL 语句创建、管理触发器。

(3) 利用 MySQL 语句创建、管理游标。

(4) 利用 MySQL 语句创建、管理事件。

3. 实验内容及步骤

(1) 游标的创建和应用。创建存储过程 pro_cursor()，利用 loop 循环语句控制 fetch 语句来检索游标 teacher_cursor 中可用的数据：

```
mysql> delimiter //
mysql> create procedure pro_cursor()
    -> begin
    -> declare  v_tno varchar(6) default  '';
    -> declare  v_tname varchar(8)  default '';
    -> declare  teacher_cursor cursor
    ->   for  select teacherno, tname from teacher;
    -> declare continue  handler for not found  set @dovar = 1;       //定义处理程序
    -> set @dovar  = 0;
    -> open teacher_cursor;
    -> fetch_Loop:loop
    -> fetch teacher_cursor into v_tno,v_tname;
    -> if @dovar = 1 then
    -> leave fetch_Loop;
    -> else
    -> select v_tno,v_tname;
    -> end if;
    -> end loop fetch_Loop;
    -> close teacher_cursor;
    -> select @dovar;
    -> end ; //
```

```
mysql > delimiter ;
mysql > call pro_cursor();
```

(2) 创建一个触发器 cno_trigger，当更改表 course 中某门课的课程号时，同时将 score 表课程号全部更新：

```
mysql > delimiter $$
mysql > create trigger cno_trigger after update
    -> on course for each row
    -> begin
    -> update  score set courseno = new.courseno
    ->   where courseno = old.courseno;
    -> end $$
mysql > delimiter ;
# 验证触发器 cno_trigger 的功能
mysql > update  course  set   courseno  = 'c08106'
    -> where   courseno = 'c07123';
mysql > select   *   from   score   where courseno  = 'c08106';
```

(3) 创建一个触发器 stu_trigger，当删除 student 表某个人的记录时，则删除 score 表相应的成绩记录：

```
mysql > delimiter $$
mysql > create   trigger stu_trigger after delete
    -> on student for each row
    -> begin
    -> delete from score where studentno = old.studentno;
    -> end $$
mysql > delimiter ;
# 验证触发器 stu_trigger 的功能
mysql > delete   from   student
    -> where   studentno = '19123567897';
mysql > select   *   from   score
    -> where studentno = '19123567897';
```

(4) 在 teacher 表中，定义一个触发器 teach_trigger，当一个教师的信息被删除时，把该教师的编号和姓名添加到 delete_teacher 表中：

```
#  创建一个空表 delete_teacher,表由 teacherno 和 tname 两列组成
mysql > create table   delete_teacher select teacherno,tname
    -> from teacher where 1 = 0;
# 创建 teacher 表的触发器
mysql > create trigger teach_trigger
    ->   after   delete on teacher for each row
    -> insert   into   delete_teacher(teacherno,tname)
    ->   values(old.teacherno, old.tname);
# 验证触发器 teach_trigger 的功能
mysql > delete   from   teacher   where   tname = '常杉';
mysql > select * from delete_teacher;
```

(5) 事件的创建和管理。

① 创建现在立刻执行的事件 direct_1,创建一个表 test:

```
mysql > create  event  direct_1
    -> on  schedule  at now()
    -> do
    -> create table test(timeline timestamp);
```

② 创建事件 test_insert,每 15 秒插入一条记录到数据表 test:

```
mysql > create  event  test_insert
    -> on  schedule  every 15 second
    -> do
    -> insert  into  test  values(current_timestamp);
```

③ 创建事件 startdays,要求从下周开始,每天都清空 test 表,并且在 2018 年的 09 月 08 日 12:00 时结束:

```
mysql > delimiter $$
mysql > create  event  startdays
    -> on  schedule  every  1  day
    -> starts  curdate() + interval 1 week
    -> ends  '2018 - 09 - 08 12:00:00'
    -> do
    -> begin
    ->   truncate  table  test;
    -> end $$
mysql > delimiter;
```

④ 查看 startdays 的创建信息:

```
mysql > show create event  startdays;
```

⑤ 删除事件 startdays 的代码如下:

```
mysql > drop event startdays;
```

4. 实验报告总结

(1) 触发器和事件在定义、调用时的区别。

(2) 触发器中 insert、delete 和 update 操作与临时表 old 与 new 的关系。

(3) 存储过程与触发器的联系和区别。

实验 11　并发事务和锁机制

1. 实验目的

(1) 掌握事务和锁的基本概念。

(2) 掌握事务的定义、管理及利用事务进行数据处理的过程。

(3) 了解 MySQL 的并发控制及锁的管理等相关技术。

2. 实验预习与准备

预习主教材第 9 章:

(1) 事务的特点和分类。

(2) 事务的启动、保存、提交、回滚操作语句。

(3) 定义隔离级别实现事务访问资源和数据的隔离。

(4) 锁的类型和管理。

3. 实验内容及步骤

(1) 创建存储过程 auto_delete，利用事务删除课程号为 c05109 的表记录，然后回滚：

```
mysql > delimiter //
mysql > set @@autocommit = 0;
    -> create procedure  auto_delete()
    -> begin
    -> start transaction;
    -> delete from course where  courseno = 'c05109';
    -> select * from course where  courseno = 'c05109';
    -> rollback;
    -> select * from  course where courseno = 'c05109';
    -> end//
#调用存储过程 auto_delete()
mysql > call auto_delete();
```

(2) 创建存储过程 tran_update，启用事务将课程表 course 中课程号为 c05109 的课程名称改为"MySQL 数据库"，并提交该事务：

```
mysql > delimiter //
mysql > create procedure  tran_update()
    -> begin
    -> start transaction;
    -> update course set cname = 'MySQL 数据库'
    -> where courseno = 'c05109';
    -> commit;
    -> select * from  course where courseno = 'c05109';
    -> end//
mysql > delimiter ;
#调用存储过程 tran_update()
mysql > call tran_update();
```

(3) 定义事务，向 course 表中添加一条记录，并设置保存点。然后再删除该记录，并回滚到事务的保存点，提交事务：

```
mysql > delimiter //
mysql > create procedure tran_save()
    -> begin
    -> start transaction;
    -> insert into course
    -> values('c05179','UML 建模','选修',48,12,7);
    -> savepoint spcno1;
    -> delete from course
    -> where courseno = 'c05179';
    -> rollback work  to  savepoint  spcno1;
    -> select * from  course where courseno = 'c05179';
    -> commit ;
```

```
        -> end //
    mysql> delimiter ;
#调用存储过程 tran_save()
    mysql> call tran_save();
```

(4) 创建存储过程 tran_student，启用事务实现在 student 表上进行查询、插入、更新和删除数据操作：

```
mysql> delimiter //
mysql> create procedure  tran_student()
    -> begin
    -> start transaction;
    -> select studentno,sname
    -> from student
    -> where  studentno = '19123567897';
    -> insert into student  values('1928261007','孙释成',
    ->  '女','1999-09-19',779,'1318909876','sdch@163.com');
    ->  update student  set sname = '孙释南'
    -> where studentno = '19123567897';
    -> commit ;
    -> end //
mysql> delimiter ;
mysql> call tran_student();
mysql> select studentno,sname,birthdate
    -> from  student where sname like '孙%';
```

(5) 锁的管理。

① 查看系统变量@@tx_isolation 中存储事务的隔离级：

```
mysql> select @@tx_isolation
```

② 在 student 表上设置一个只读锁定：

```
mysql> lock tables student read;
```

③ 在 student 表上设置一个写锁定：

```
mysql> lock tables student write;
```

④ 解锁表：

```
mysql> unlock tables;
```

4. 实验报告总结

(1) 并发数据访问引发的问题及解决方案。

(2) 回滚和检查点的作用。

(3) 如何避免出现死锁。

实验 12　MySQL 的安全管理

1. 实验目的

(1) 了解 MySQL 的权限系统的工作原理。

(2) 掌握 MySQL 账户安全管理的基本操作。

(3) 掌握 MySQL 权限安全管理的基本操作。

2. 实验预习与准备

预习主教材第 10 章：

(1) MySQL 的权限机制与权限管理。

(2) 账户管理的基本操作。

(3) 权限管理的基本操作。

3. 实验内容及步骤

(1) MySQL 命令的使用。

① 使用 root 用户登录到本地 MySQL 服务器的 teaching 库中：

```
mysql -uroot -p -hlocalhost  teaching
```

② 使用 root 用户登录到本地 MySQL 服务器的 teaching 库中，执行一条查询语句：

```
mysql -uroot -p -hlocalhost teaching -e "desc student;"
```

③ 使用 mysqladmin 将 root 用户的密码修改为"123457"：

```
mysqladmin -u root -p password "123457"
```

④ 使用 update 语句将 root 用户的密码修改为"pwd1357"：

```
mysql> update mysql.user set password = password("pwd1357")
    -> where  user = "root" and host = "localhost";
```

⑤ 使用 set 语句将 root 用户的密码修改为"pwd258"：

```
mysql> set  password = password("pwd258");
```

(2) 普通用户的管理。

① 使用 create user 创建一个用户，用户名是 jef，密码是 mypass，主机名是 localhost：

```
mysql> create user 'jef'@'localhost' identified  by 'myword';
```

② 使用 drop user 删除用户 'jeff'@'localhost'：

```
mysql> drop user 'jeff'@'localhost';
```

③ 使用 grant 语句创建一个新的用户 usertest，密码为 testpwd。用户 usertest 对所有的数据有查询和更新权限，并授予对所有数据表的 select 和 update 权限：

```
mysql> grant select ,update  on  *.* to 'usertest'@'localhost'
    -> identified by 'testpwd';
mysql> select Host,User,Select_priv,Update_priv,from mysql.user
    -> where user = 'usertest';
```

④ 使用 insert 创建一个新账户，其用户名称为 cust，主机名称为 localhost，密码为 customer1：

```
mysql> insert into  user (Host,User,Password)
```

```
    -> values('localhost','cust',password('customer1'));
```

⑤ 使用 delete 删除用户 'cust'@'localhost'：

```
mysql> delete from mysql.user
    -> where host = 'localhost' and user = 'cust';
```

⑥ 使用 set 语句将 usertest 用户的密码修改为"newpwdone"：

```
mysql> set password for 'usertest'@'localhost' = password("newpwdone");
```

⑦ 使用 update 语句将 usertest 用户的密码修改为"newpwdtwo"：

```
mysql> update   mysql.user set Password = password("newpwdtwo")
    -> where  User = "usertest" and Host = "localhost";
```

⑧ 使用 grant 语句将 usertest 用户的密码修改为"newpwd3"：

```
mysql> grant usage on  *.* to 'usertest'@'localhost'
    -> identified by 'newpwd3';
```

（3）权限管理。

① 使用 grant 语句创建一个新的用户 grantuser，密码为 grantpwd。用户 grantuser 对所有的数据有查询、插入权限，并授予 grant 权限：

```
mysql> grant select ,insert on *.* to 'grantuser'@'localhost'
    -> identified by 'grantpwd'
    -> with grant option;
mysql> select Host,User,Select_priv,Insert_priv, grant_priv
    -> from  mysql.user
    -> where user = 'grantuser';
```

② 使用 revoke 语句取消用户 grantuser 的更新权限：

```
mysql>  revoke update on *.* from 'grantuser'@'localhost';
mysql>  select Host,User,Select_priv,Update_priv,Grant_priv
    -> from  mysql.user
    -> where user = 'usertest';
```

③ 使用 show grants 语句查询用户 grantuser 的权限信息：

```
mysql> show grants for 'grantuser'@'localhost';
```

④ 收回 grantuser 用户对 teaching 数据库中 course 表的 update 权限：

```
mysql> revoke update on teaching. course
    -> from grantuser@localhost;
```

⑤ 授予 grantuser 每小时可以发出的查询数 18 次，每小时可以连接数据库 5 次，每小时可以发出的更新数为 15 次的权限：

```
mysql> grant all on *.* to grantuser@localhost
    -> identified by 'grantpass'
    -> with max_queries_per_hour 18
```

```
    -> max_updateS_per_hour 15
    -> max_connections_per_hour 10;
```

⑥ 使用 revoke 语句收回 grantuser 用户的所有权限，包括 grant 权限：

```
mysql> revoke all privileges,grant option
    -> from grantuser@localhost;
```

4. 实验报告总结

(1) MySQL 系统的安全机制的优缺点。

(2) MySQL 数据库安全管理中要注意的问题。

(3) 授予权限和撤销权限的关系。

实验 13　备份和恢复

1. 实验目的

(1) 了解备份和恢复数据库的备份策略的选择。

(2) 掌握备份数据库的基本操作。

(3) 掌握恢复数据库的基本操作。

(4) 掌握表的导入导出的基本操作。

2. 实验预习与准备

预习主教材第 11 章：

(1) 备份和还原的时机与类型。

(2) 备份数据库的方法与步骤。

(3) 还原数据库的方法与步骤。

(4) 表数据的导入导出步骤。

3. 实验内容及步骤

(1) 使用 mysqldump 备份数据库和表。

① 使用 mysqldump 命令备份数据库 teaching 中的所有表：

```
mysqldump -u root -p teaching > f:/backup/teaching_20180101.sql
```

② 备份 teaching 数据库中的 teacher 表：

```
mysqldump -u root -p teaching teacher > f:/backup/teacher_20180101.sql
```

③ 使用 mysqldump 备份 teaching 和 mysqltest 数据库：

```
mysqldump -u root -p --databases  teaching  mysqltest
        > f:\backup \teach_testdb_20180101.sql
```

④ 使用 mysqldump 备份服务器中的所有数据库：

```
mysqldump  -u root -p --all-databases > f:/backup/alldb.sql
```

⑤ 使用 mysqlhotcopy 备份 mysqltest 数据库到/usr/backup 目录下：

```
mysqlhotcopy  -u root -p mysqltest  /usr/backup
```

(2) 数据恢复。

① 使用 mysql 命令将 f:\backup\teaching_20180101.sql 文件中的备份导入到数据库中：

```
mysql -u root -p teaching < f:/backup/teaching_20180101.sql
```

② 使用 root 登录到服务器，使用 source 导入本地的备份文件 teaching_201800101.sql：

```
mysql> use teaching;
mysql> source f:/backup/teaching_201800101.sql
```

③ 从 mysqlhotcopy 拷贝的备份恢复数据库：

```
cp -r /usr/backup/mysqltest  usr/local/mysql/data
```

(3) 数据库迁移。将 www.a123.com 主机上的 MySQL 数据库全部迁移到 www.b123.com 主机上。在 www.abc123.com 主机上执行的命令如下：

```
mysqldump -h www.a123.com -uroot -ppassword  dbname |
mysql -hwww.b123.com -uroot -ppassword
```

(4) 数据的导入导出。

① 使用 select… into outfile 将 teaching 数据库中的 student 表中的记录导出到文本文件：

```
mysql> select *  from teaching.student into outfile 'f:/student.txt';
```

② 使用 select …into outfile 将 teaching 数据库中的 student 表中的记录导出到文本文件，使用 fields 选项和 lines 选项，要求字段之间使用逗号“，”间隔，所有字段值用双引号括起来，定义转义字符定义为单引号“\'”：

```
mysql> select * from teaching.student into outfile 'f:/student01.txt'
    -> fields
    ->   terminated by ','
    ->     enclosed by '\"'
    ->     escaped by '\''
    -> lines
    ->   terminated by '\r\n';
```

③ 使用 select …into outfile 将 teaching 数据库中的 student 表中的记录导出到文本文件，使用 lines 选项，要求每行记录以字符串“>”开始，以“< end >”字符串结尾。

```
mysql> select * from teaching.student into outfile  "f:/student02.txt"
    -> lines
    ->   starting by '> '
    ->   terminated by '< end >';
```

④ 使用 mysqldump 将 teaching 数据库中的 student 表中的记录导出到文本文件：

```
mysqldump - t f:/ teaching student -u root -p
```

⑤ 使用 mysqldump 命令将 teaching 数据库中的 student 表中的记录导出到文本文件，使用 FIELDS 选项，要求字段之间使用逗号“,”间隔，所有字符类型字段值用双引号括起来，定义转义字符定义为问号“?”，每行记录以回车换行符“\r\n”结尾：

```
mysqldump -t f:\backup teaching student -u root -p
 --fields-terminated-by=,
 --fields-optionally-enclosed-by=\"
 --fields-escaped-by=?
 --lines-terminated-by=\r\n
```

⑥ 使用 MySQL 语句导出 teaching 数据库中 student 表中的记录到文本文件：

```
mysql -u root -p
   --execute="select * from  student; " teaching  >f:\student3.txt
```

⑦ 使用 load data 命令将 f:\student.txt 文件中的数据导入到 teaching 数据库中的 student 表：

```
mysql> Load data infile 'f:/student.txt' into rable teaching.student;
```

4. 实验报告总结

(1) MySQL 数据库的备份策略选择。

(2) 还原数据库需要注意的问题。

实验 14　性能优化和日志文件管理

1. 实验目的

(1) 掌握 MySQL 性能优化的概念和原理。

(2) 掌握利用日志文件的分类和功能。

(3) 学会利用日志文件优化数据库的操作。

2. 实验预习与准备

预习主教材第 12 章和第 13 章。

(1) MySQL 性能优化的基本常用方法。

(2) 优化查询和优化数据库结构方法。

(3) 日志文件的启用、查看、删除等常见操作。

3. 实验内容及步骤

(1) MySQL 性能优化参数的查看和设置。

① 查询慢查询次数：

```
mysql> show status like 'Slow_queries';
```

② 查询 MySQL 数据库的查询缓冲区的大小：

```
mysql> show variables like 'query_cache_size ';
```

③ 设置查询缓冲区的系统变量：

```
mysql> set @@global.query_cache_limit=67108864;
```

④ 关闭查询缓冲区：

```
mysql> use  MYSQL;
mysql> set @@Query_cache_type = 0;
```

（2）优化查询。

① 使用 explain 语句分析一个查询语句：

```
mysql> explain  select *  from  score;
```

② 使用 describe 命令分析查询语句：

```
mysql> describe select * from course;
```

③ 分析索引对查询速度的影响，在未使用索引时的查询情况：

```
mysql> explain select * from student  where  phone = '13112345677';
```

④ 使用 explain 语句执行查询命令，应用 like 关键字，且匹配字符串中含有百分号"%"：

```
mysql> explain select * from student  where  sname  like '刘%'\G
mysql> explain select * from student where sname like '%梅'\G
```

⑤ 通过 explain 来分析应用多列索引的命令。score 表中有索引 studentno+courseno，分别用这两个字段进行查询分析：

```
mysql> explain  select * from score where studentno = '19%'\G
mysql> explain  select * from score where courseno = 'c03%'\G
```

（3）优化数据库结构。

① 分析 student 表的运行情况，先使用 show index 语句来查看索引的散列程度，然后可以使用 analyze table 进行修复：

```
mysql> show index from student\G;
mysql> analyze table student ;
```

② 检查表 student 的运行情况：

```
mysql> Check Table student;
```

③ 利用 optimize table 语句优化 student 表：

```
mysql> optimize table student;
```

④ 将查询时间超过 2 秒的查询作为慢查询：

```
mysql> set long_query_time = 2;
mysql> show variables like 'long%';
```

⑤ 查询 student 表高速缓存运行中的反应结果：

```
mysql> set @@global.query_cache_size = 1;
#使用查询高速缓存运行结果
```

```
mysql> select sql_cache * from teacher;
#未使用查询高速缓存运行结果
mysql> select sql_no_cache * from teacher;
```

(4) 错误日志的管理。

① 使用记事本查看 MySQL 错误日志:

```
mysql> show variables like 'log_error';          //查找到错误文件用记事本打开
```

② 使用 mysqladmin 命令来开启新的错误日志:

```
C:\> mysqladmin -u root -p flush -logs
```

③ 删除错误日志:

```
mysql> flush logs;
```

(5) 二进制日志管理。

① 使用 show variables 语句查询日志设置:

```
mysql> show variables like 'log_%';
```

② 使用 show binary logs 查看二进制日志文件的个数及文件名:

```
mysql>  show binary logs;
```

③ 使用 mysqlbinlog 查看二进制日志 PGabc-bin.0001:

```
C:\Users\Administrator
  >mysqlbinlog C:\ProgramData\MySQL\MySQL Server5.7 \Data\PGabc-bin.0001
```

④ 使用 purge master logs 删除创建时间比 PGabc-bin.0003 早的所有日志文件:

```
mysql>  purge master logs TO 'PGabc-bin.0003';
```

⑤ 使用 purge master logs 删除 2018 年 6 月 12 日前创建的所有日志文件:

```
mysql> purge master logs before '20180612';
```

⑥ 使用二进制日志恢复数据库。使用 mysqlbinlog 恢复 MySQL 数据库到 2018 年 6 月 12 日 17:00:00 时的状态:

```
C:\> mysqlbinlog - stop-date="2018-06-12 17:00:00"
      C:\Documents and Settings\All Users\MySQL
      \MySQL Server 5.7\Data\ PGabc-bin.0018 -uroot  -p
 Enter password: ******
```

4. 实验报告总结

(1) MySQL 性能优化的优点。

(2) MySQL 日志管理的用途。

实验 15 PHP 管理 MySQL 数据表

1. 实验目的

(1) 掌握 PHP 语言的优势和基本运行操作。

(2) 掌握搭建 MySQL+PHP 集成环境的步骤。

(3) 掌握使用 PHP 操作数据库的基本操作。

2. 实验预习与准备

预习主教材第 14 章：

(1) PHP 语言的特点和工作管理。

(2) 创建和运行 PHP 程序的步骤。

(3) 利用 PHP 连接 MySQL 数据库的方法。

3. 实验内容及步骤

(1) 认识和了解 PHP 语言的基本操作。

① 表达式的输出：

```
<?php
echo 5 + 5;                                 //10
echo '<br>';
echo 6 - 4;                                 //2
echo '<br>';
echo 3 * 4;                                 //12
echo '<br>';
echo 3/2;                                   //1.5
echo '<br>';
echo 5 % 7;                                 //5
echo '<br>';
echo 2 ** 3;                                //8 (PHP7.1.6)
?>
```

② PHP 语言中 if…else 语句的认识：

```
<?php
   $score = 85;
  if( $score >= 90 ){
 echo '优秀';
  }elseif( $score >= 80 ){
 echo '良好';
  }elseif( $score >= 70 ){
 echo '中等';
  }elseif( $score >= 60 ){
 echo '及格';
  }else{
 echo '不及格';
  }
?>
```

③ PHP 语言中 for 语句的认识：

```
<?php
    $sum = 0;                               //用于保存1~100的奇数和
   for( $i = 1; $i <= 100; ++ $i){
  if( $i % 2 == 0){                         //若为偶数,则不累加
     continue;                              //结束本次循环
    }
    $sum += $i;                             //累加奇数
```

```
    }
  echo '$sum = '.$sum;                        //$sum = 2500
?>
```

④ PHP 语言中 while 语句的认识：

```
<?php
   $i = 7;
  while( $i>0 ){
 echo $i--;                                   //循环输出 7654321
   }
?>
```

(2) 利用 PHP 语言查询数据表 teacher 中的数据：

```
<?php
 $conn1 = mysqli_connect("localhost","root","123456");
 $select = mysqli_select_db( $conn1,"teaching");          //连接服务器中的 teaching
if( $select)
    {
    header("Content-Type:text/html;charset = gb2312");    //设置字符集
  echo "数据库连接成功!" ;                                //判断是否连接成功
    }
 $query = "select * from  teacher";
 $result = mysqli_query( $conn1, $query) or die("查询失败!".mysqli_error());
echo mysqli_affected_rows( $conn1);
?>
```

(3) 利用 PHP 语言更新数据表 student 中的数据：

```
<?php
 $conn1 = mysqli_connect("localhost","root","123456");    //连接 MySQL 服务器
 $select = mysqli_select_db( $conn1,"teaching");          //连接服务器中的 teaching
if( $select)
    {
    header("Content-Type:text/html;charset = gb2312");    //设置字符集
  echo "数据库连接成功!" ;                                //判断是否连接成功
    }
 $sqldelete = " update score  set final = 89 where studentno = '19122111208' and courseno =
'c06127'";
mysqli_query( $conn1, $sqldelete);
?>
#检测更新数据
mysql> select * from  score
   -> where studentno = '19122111208' and courseno = 'c06127';
```

(4) 查询课程号为 c05109 的成绩信息，并利用输出命令 echo 和 print_r()函数两种方式输出：

```
<?php
 $conn1 = mysqli_connect("localhost","root","123456");    //连接 MySQL 数据库服务器
 $select = mysqli_select_db( $conn1,"teaching");          //连接服务器中的 teaching
if( $select){
   header("Content-Type:text/html;charset = gb2312");     //设置字符集
   echo "数据库连接成功!" ;                               //判断是否连接成功
```

```
}
$ sql = "select * from score where courseno = 'c05109';";
if ( $ result = mysqli_query( $ conn1, $ sql))
    {
      while ( $ row = mysqli_fetch_assoc( $ result))
        {
          echo  "< br />";
          echo  "echo 格式: "."< br />";
          echo  "学号      ". $ row['studentno'];
          echo  "  课程号 ". $ row['courseno'];
          echo  "  平时成绩". $ row['daily'];
          echo  "  期末成绩". $ row['final']."< br/>";
          echo  "print_r()函数格式: "."< br />";
          print_r( $ row);
        }
      mysqli_free_result( $ result);                    //释放内存
    }
   mysqli_close( $ conn1);                              //关闭连接对象
?>
```

4. 实验报告总结

(1) 配置 PHP+MySQL 集成环境的注意事项。

(2) 利用 PHP 连接 MySQL 数据库的步骤。

实验 16 综合练习

1. 实验目的

(1) 使用可视化工具 phpMyAdmin 的基本操作。

(2) MySQL 数据库和表的常用操作。

(3) MySQL 数据库对象的常用操作。

(4) MySQL 数据库的备份、还原,数据导入导出、日志管理、事务和锁的可视化操作。

2. 实验预习与准备

预习本书第 17 章:

(1) 可视化工具 phpMyAdmin 的安装和配置。

(2) 可视化工具 phpMyAdmin 的功能。

(3) phpMyAdmin 与 MySQL 数据库连接的步骤。

3. 实验内容及步骤

(1) 数据库的基本操作。通过 phpMyAdmin 登录 MySQL 后,phpMyAdmin 的主界面如图 18-14 所示。

① 创建数据库。单击左边窗口的“新建”按钮,进入如图 18-15 所示的创建数据库页面。首先在文本框中输入数据库的名称 carsales_info,然后在下拉列表框中选择所要使用的编码,一般选择 gb2312_Chinese_ci 简体中文编码格式,单击“创建”按钮,即可创建数据库 carsales_info。成功创建数据库后,carsales_info 数据库将显示在左侧窗口的界面。

说明:*在右侧界面中可以对该数据库进行相关操作,如数据库、SQL、账户、导出、导入、设置、二进制日志、复制、变量、字符集、插件等,单击相应的超链接进入相应的操作界面。还可以观察到数据库服务器、网站服务器和 phpMyAdmin 的参数信息。*

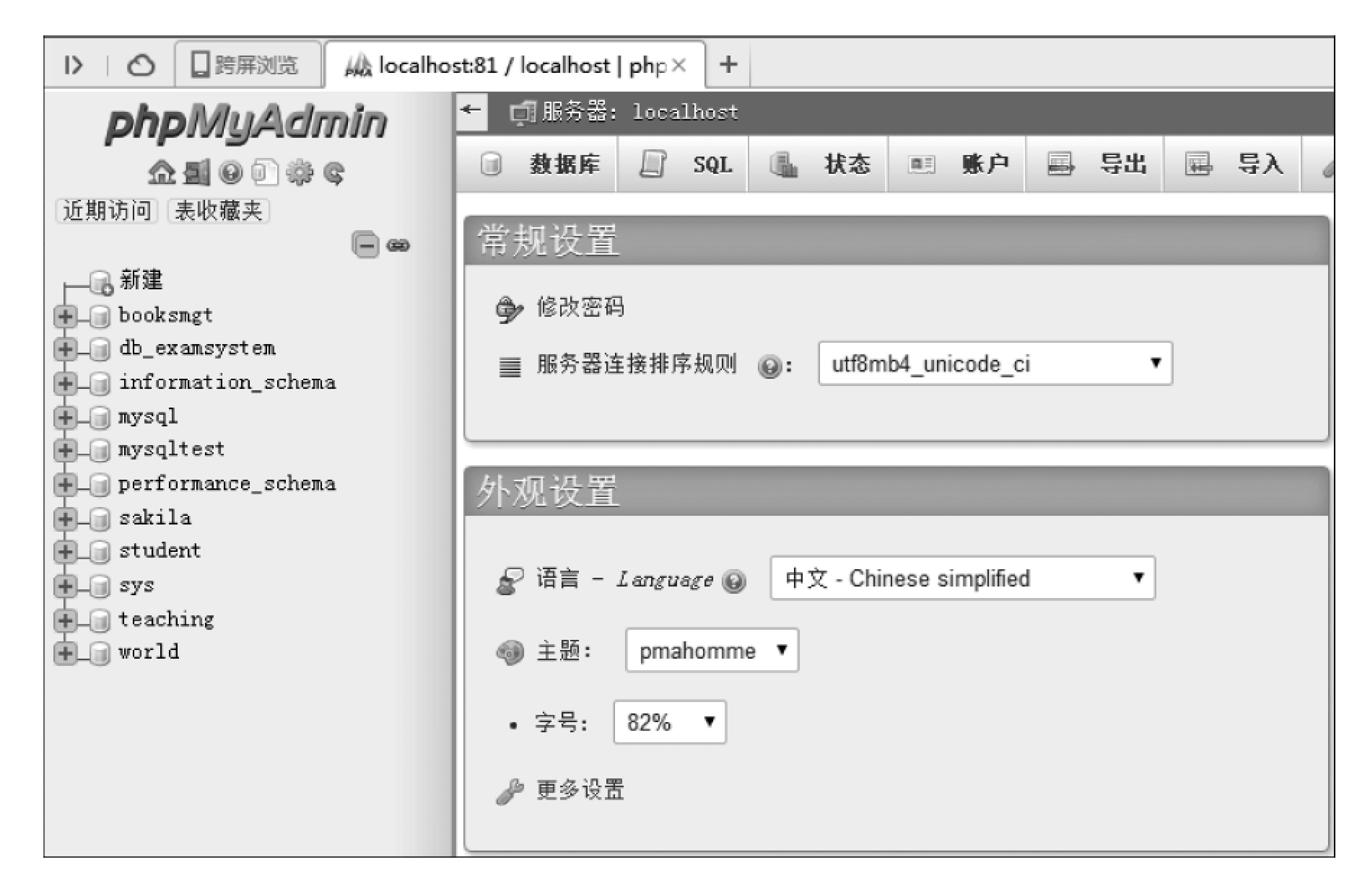


图 18-14　phpMyAdmin 的主界面



图 18-15　创建数据库

② 删除数据库。单击"数据库"超链接,进入如图 18-16 所示的界面中。要删除某个数据库,如 student,首先在数据库左侧选择该数据库,然后单击下面的"删除"按钮,进入如图 18-17 所示的对话框中,单击"确定"按钮,即可成功删除指定的数据库。



图 18-16　修改数据库



图 18-17　删除数据库

(2) 数据表的基本操作。选择新建的数据库 carsales_info，在该数据库中可以创建、修改和删除数据表。

① 创建数据表。创建数据库 carsales_info 后，进入如图 18-18 所示的页面。在该操作页面中输入数据表的名称(如 carsinfo)和字段数，然后单击"执行"按钮，即可进入如图 18-19 所示的创建数据表 carsinfo 的页面。依次按照预先设计好的数据表结构输入字段名、选择字段类型、输入字段长度等，还可以按照如图 18-20 所示创建主键索引。输入完毕，单击"执行"按钮，即可完成数据表 carsinfo 的结构创建，如图 18-21 所示。



图 18-18　新建数据表



图 18-19　添加字段

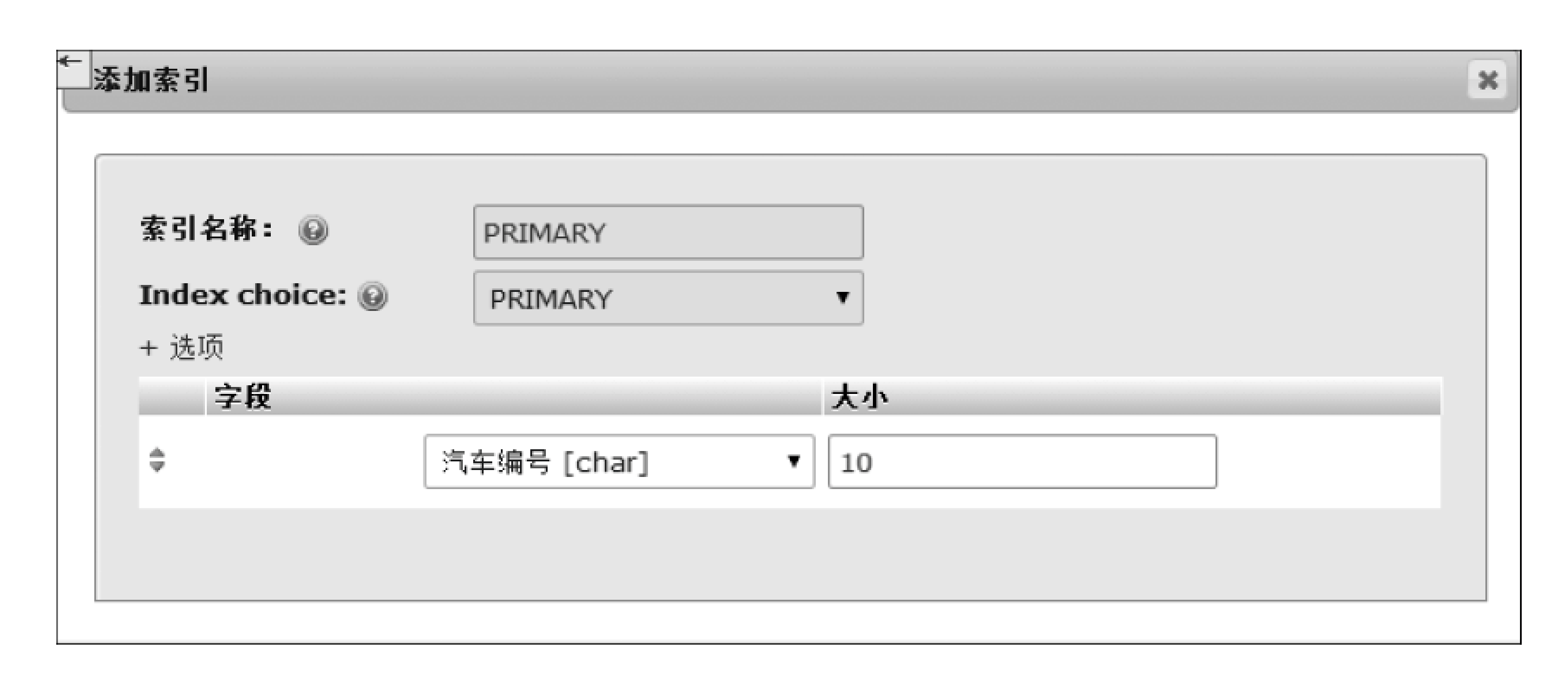


图 18-20　创建主键



图 18-21 完成表结构创建

成功创建数据表 carsinfo 后，将显示数据表结构界面。在表单中对各个字段的详细信息进行录入，包括字段名、数据类型、长度/值、排序规则、是否为空、主键等，以完成对表结构的详细设置。

② 添加记录。在图 18-21 所示的数据表结构页面中单击“插入”按钮，依次输入各字段的值，如图 18-22 所示，输入完毕单击“执行”按钮，完成一行记录的输入。按照此方式可以输入余下的记录，输入完毕，单击“浏览”按钮，如图 18-23 所示。



图 18-22 添加记录

③ 修改记录。如果需要修改记录，可以单击图 18-23 所示的每一行记录的“编辑”按钮，进入如图 18-24 所示的修改记录对话框，修改相关数据项。修改完毕，单击“执行”按钮即可。此时还可以单击此页下面的“预览 SQL 语句”按钮，出现如图 18-25 所示的对话框。

④ 按照此方法，可以创建其他表，并输入相关记录。也可通过 MySQL 客户端窗口执行相关操作。数据库 carsales_info 的 6 个表的结构和数据录入完成后，界面如图 18-26 所示。



图 18-23　完成记录添加

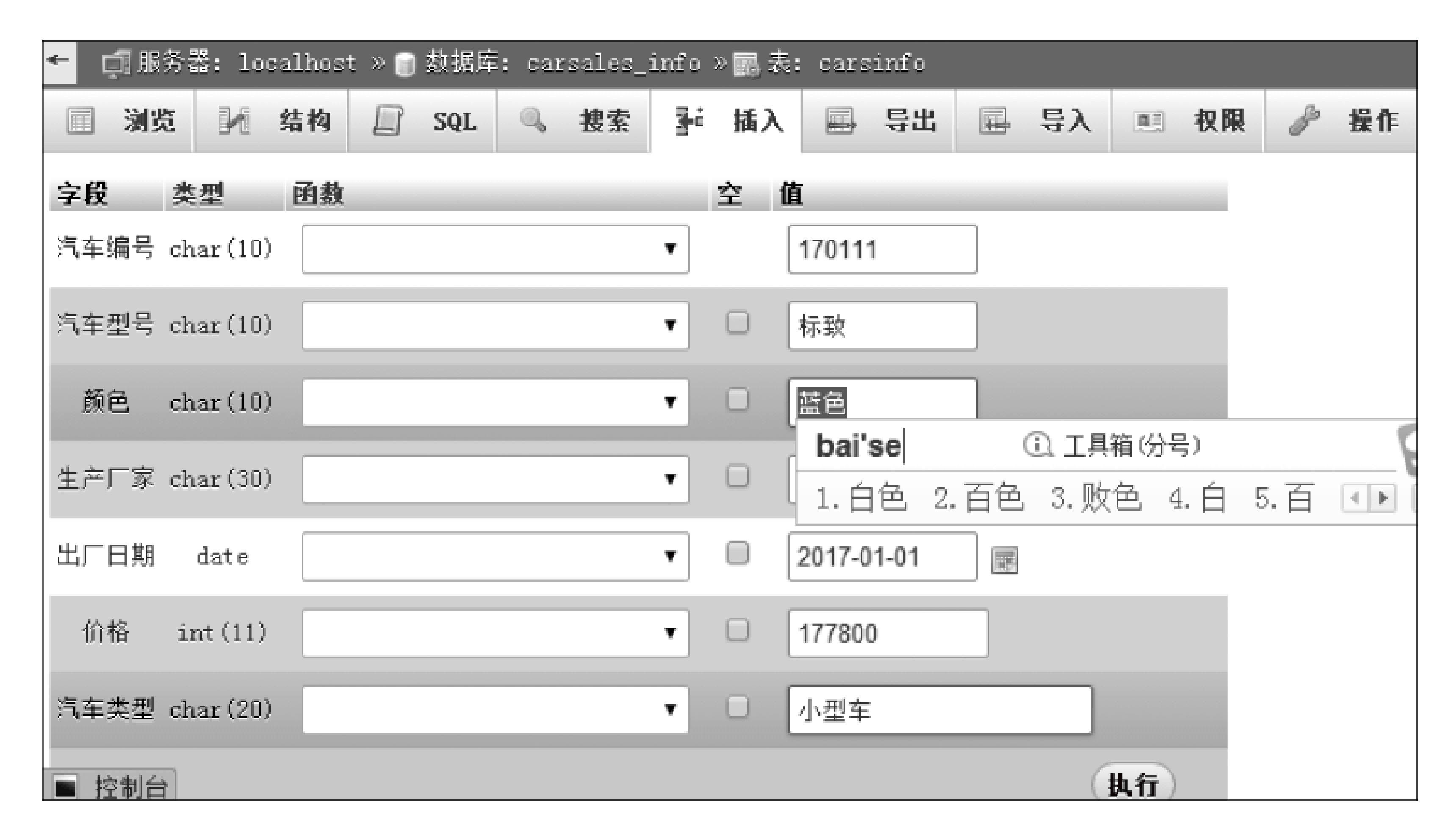


图 18-24　修改记录



图 18-25　修改记录的 SQL 语句



图 18-26　创建完成其他表

⑤ 创建外键约束。浏览表 business，单击“结构”超链接，再单击“关联视图”，在此页面中输入外键名 fk_emp，添加“员工编号”字段，按照图 18-27 所示的内容选择数据库 carsales_info、表 employee、字段“员工编号”，并选择 ON DELETE 和 ON UPDATE 后的选项为 RESTRICT。单击页面下方的“保存”按钮即可。



图 18-27　创建外键

外键约束创建后，设计器中的 business 和表 employee 也会自动连线，如图 18-28 所示。

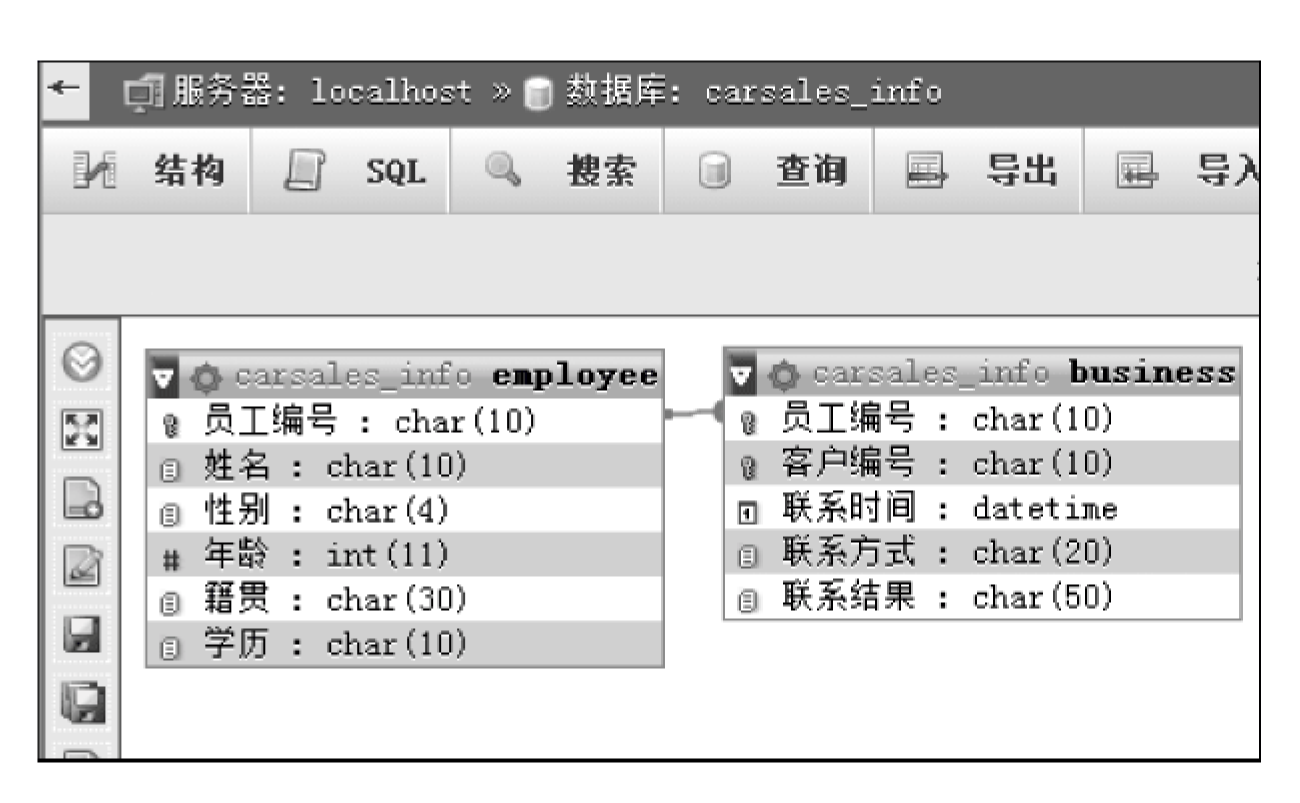


图 18-28　数据库的设计器

⑥ 修改表结构。数据表被创建后，可以在图 18-26 所示的数据表结构页面中修改表结构，可以执行添加、删除、修改字段的数据类型或者字段的长度/值等操作。也可以创建、修改索引字段。

例如，单击表 employee 的“结构”超链接，进入如图 18-29 所示的修改表结构的对话框。单击“员工编号”字段行的“修改”按钮，在弹出的对话框中将字段长度 20 修改为 10，单击“预览 SQL 语句”按钮，如图 18-30 所示，单击“关闭”按钮之后，再单击“保存”按钮，即可完成数据表的结构的修改。

⑦ 删除数据表。要删除某个数据表，首先选择数据库，在图 18-26 所示的数据库中选择数据表，单击“删除”按钮，然后在弹出的对话框中单击“确定”按钮即可成功删除指定的数据表。



图 18-29　修改表结构



图 18-30　修改数据表结构

(3) 利用SQL语句管理数据记录。在phpMyAdmin主界面中选择一个数据表carsinfo,单击"SQL"超链接,可以打开SQL语句编辑区。在编辑区输入完整的SQL语句,来实现数据的查询、添加、修改和删除操作,如图18-31所示。



图 18-31　利用SQL语句管理表数据

① 使用SQL语句插入数据。在SQL语句编辑区应用insert语句向数据表carsinfo中插入数据后,如图18-32所示。单击"执行"按钮,向数据表中插入2条数据,如图18-33所示。如果提交的SQL语句有错误,系统会给出一个警告,提示用户进行修改。



图 18-32　使用 SQL 语句向数据表中插入数据



图 18-33　插入数据成功

说明：还可以利用其右侧的属性列表来选择要操作的列，只要选中要添加的列，双击其选项或者单击“<<”按钮添加列名称。

② 使用 SQL 语句修改数据。在 SQL 语句编辑区应用 update 语句修改数据信息，将汽车编号为 170117 的汽车型号改为“标致 708”，颜色为金色，添加的 SQL 语句如图 18-34 所示。单击“执行”按钮，数据修改成功。

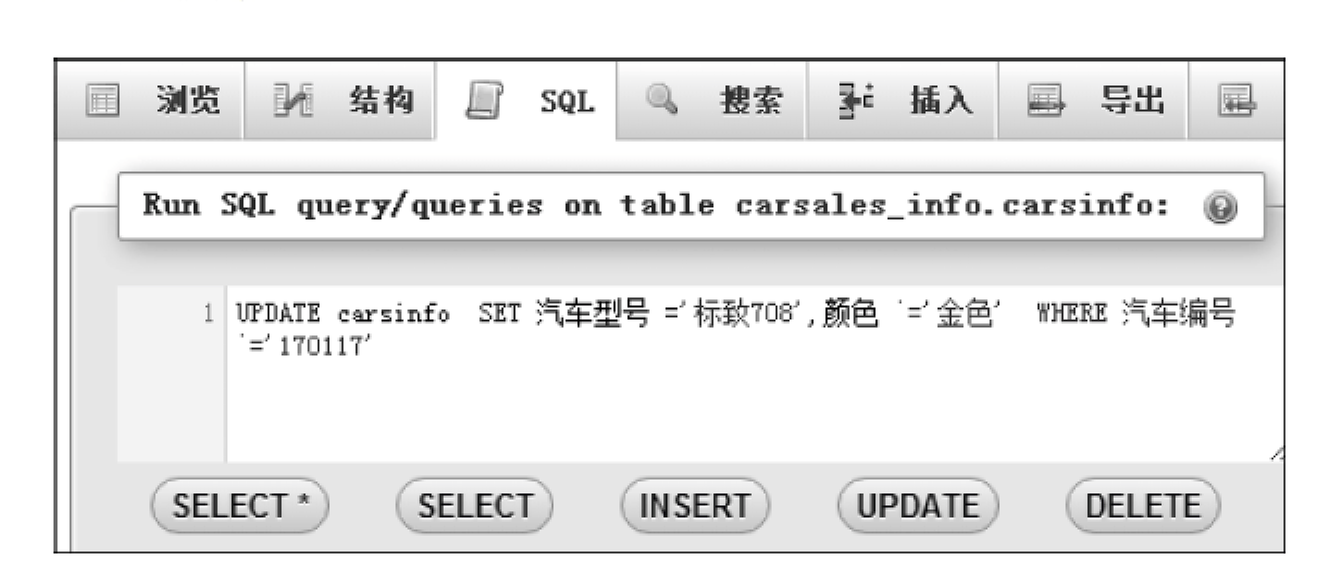


图 18-34　使用 SQL 语句向数据表中修改数据

③ 使用 SQL 语句查询数据。在 SQL 语句编辑区应用 select 语句检索指定条件的数据信息，将汽车编号前 4 位为 1701 的数据全部显示出来，添加的 SQL 语句如图 18-35 所示。单击“执行”按钮，该语句的实现结果如图 18-36 所示。



图 18-35　添加查询数据信息的 SQL 语句



图 18-36　查询指定条件的数据信息

说明:此 SQL 编辑区可以进行整个表的简单查询外,还可以执行复杂的条件查询,如使用 where 子句提交 like、order by、group by 等条件查询子句。也可以进行多表连接查询、子查询等。

④ 使用 SQL 语句删除数据。在 SQL 语句编辑区应用 delete 语句检索指定条件的数据或全部数据信息,删除名称为“计非竹”的员工信息,添加的 SQL 语句如图 18-37 所示。

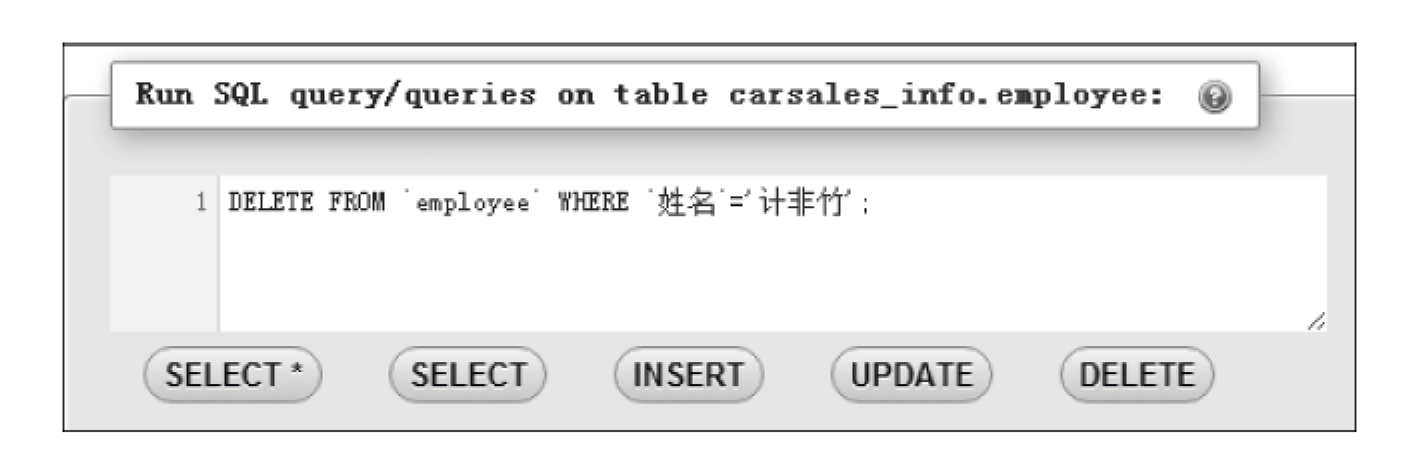


图 18-37　删除指定数据信息的 SQL 语句

单击“执行”按钮,弹出确认删除操作对话框,单击“确定”按钮,执行数据表中指定条件的删除操作。如果 delete 语句后面没有 where 条件值,那么将删除指定数据表中的全部数据。

(4) 搜索数据。选择某个数据表后,单击“搜索”超链接,进入搜索页面。在这个页面中,可以在选择字段的列表框中指定搜索条件,如图 18-38 所示,可以对表 carsinfo 的记录按条件进行查询。选择查询的条件后,单击“执行”按钮即可。

(5) 数据的导入导出。使用 phpMyAdmin 软件导入导出数据,可以实现对数据库的结构和数据的导入导出,包括导入导出多种格式的文件,本质上这是一种转换数据格式的复



图 18-38　搜索查询

制。导入导出可以是在数据库和表两种模式下进行，操作过程和步骤相近。

① 导出数据表的结构和数据。选择数据表 carsinfo，单击页面中的“导出”超链接，打开导出编辑区。从下拉框中选择导出文件的格式，默认使用选项“SQL”。如图 18-39 所示，这里使用选项“CSV for MS Excel”，单击“执行”按钮，弹出如图 18-40 所示的文件下载对话框，单击“下载”按钮，将数据文件以“. csv”格式存储在指定位置。



图 18-39　导出 MySQL 表的设置界面

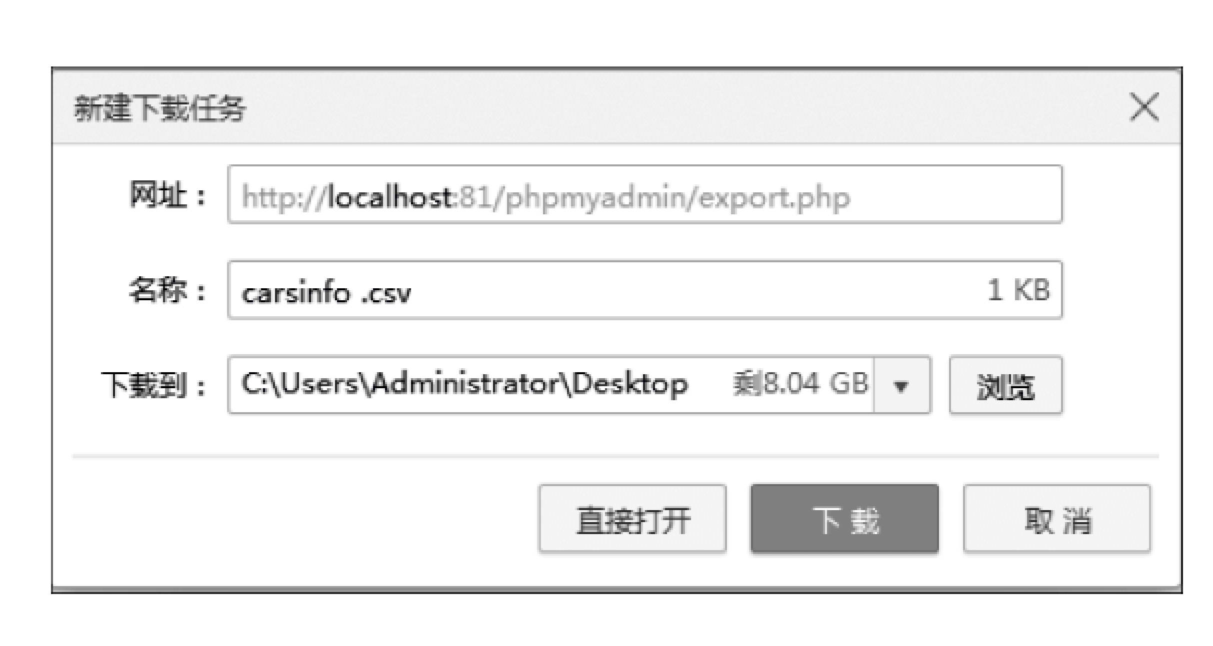


图 18-40　存储 MySQL 数据的对话框

② 导入 MySQL 数据库脚本。单击“导入”超链接，进入执行 MySQL 数据库的脚本界面，单击“选择文件”按钮查找脚本文件(如 carsinfo1. csv)所在位置，如图 18-41 所示，单击“执行”按钮，即可执行导入 MySQL 数据库脚本文件。导入的表的名称为 table7，如图 18-42 所示。此时可以浏览 table7 数据，也可以修改表名。



图 18-41　执行 MySQL 数据库脚本文件



图 18-42　成功执行导入操作

(6) 数据库的用户和权限管理。在 phpMyAdmin 图形化管理工具中，不但可以对 MySQL 数据库进行各种操作，而且可以添加服务器的新用户，并对新旧用户设置权限。

① 添加 MySQL 服务器新用户。单击 phpMyAdmin 主界面中的“账户”超链接，打开服务器用户操作界面，如图 18-43 所示。

在该页面中，单击“新增用户账户”。进入如图 18-44 所示的界面，设置用户名、密码、主机，并对新用户的权限进行设置。设置完成后，单击“执行”按钮，完成对新用户的添加操作，返回主页面，将提示新用户添加成功，如图 18-45 所示。

② 重置用户登录密码。在 phpMyAdmin 图形化管理工具中，不但可以对 MySQL 数据库进行各种操作，而且可以对用户的权限进行重新设置，同时还可以对 MySQL 服务器的登录密码进行重置。单击图 18-45 中的“修改密码”按钮，打开服务器用户操作界面，可以在此



图 18-43　服务器用户一览表

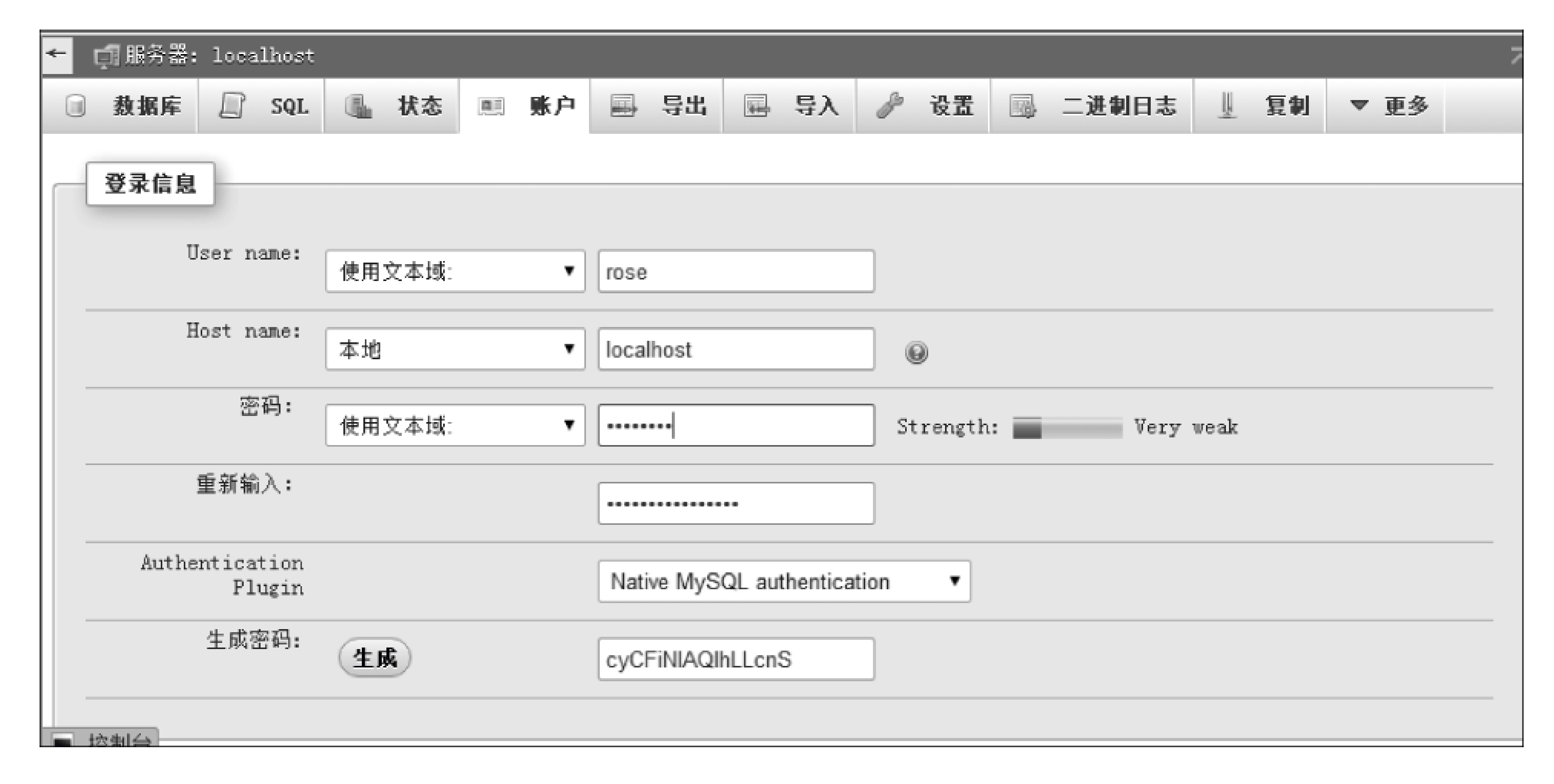


图 18-44　设置添加用户信息



图 18-45　成功添加用户信息

重新设置密码，还可以单击“生成”按钮，由系统生成密码，如图 18-46 所示。

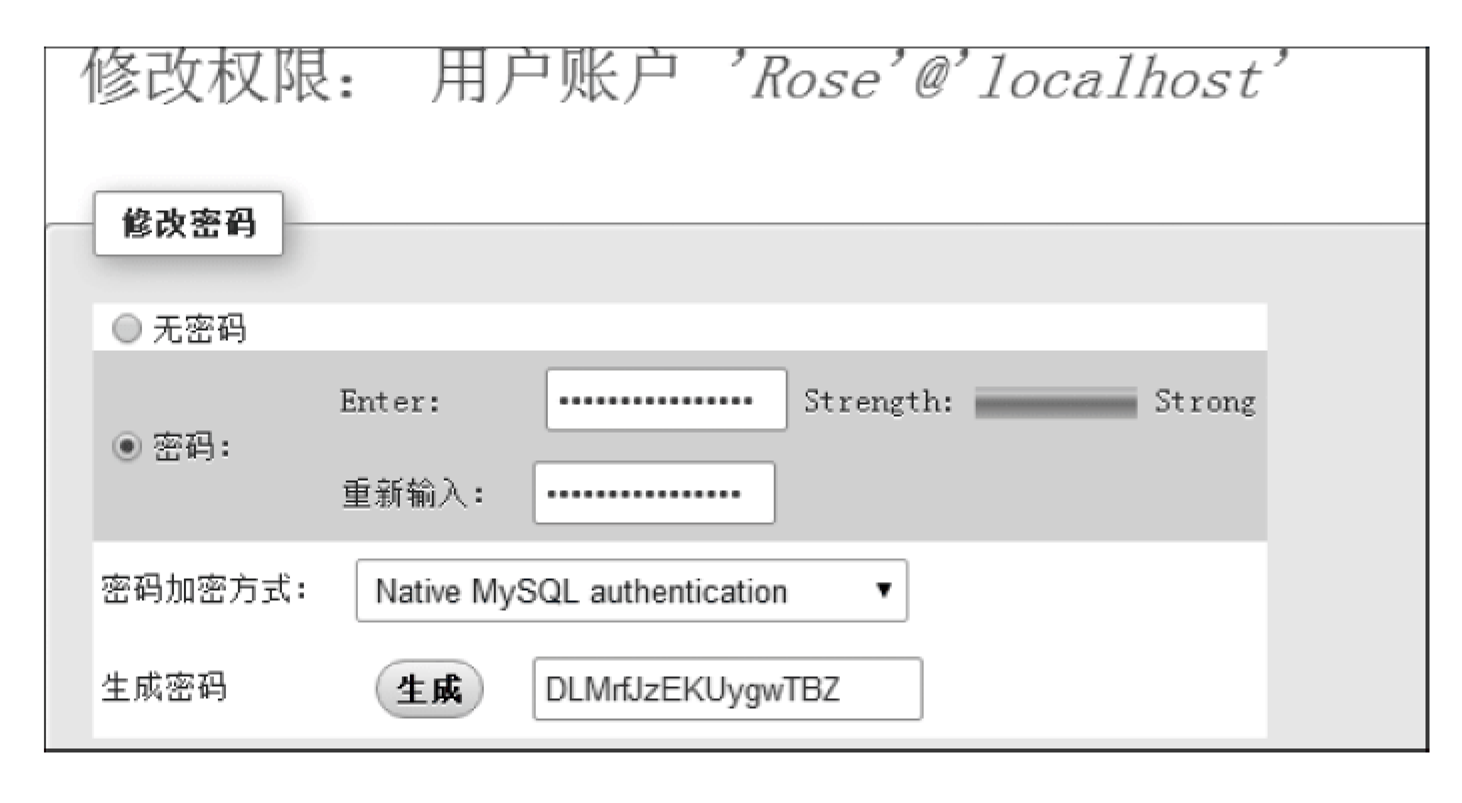


图 18-46 修改用户密码

③ 修改用户权限。在图 18-43 所示的页面中，可以对指定用户的权限进行编辑、可以添加新用户和删除指定的用户。这里选择指定的用户，单击“修改权限”超链接，对指定用户的权限进行设置，如图 18-47 所示。在此可以设置用户的权限后，单击“执行”按钮，完成对用户权限的修改操作，返回主页面。



图 18-47 修改用户权限

（7）存储过程的基本操作。存储过程是 MySQL 中常用的数据库对象，在 phpMyAdmin 图形化管理工具中存储函数、触发器和事件等数据库对象的操作过程与此类似。

① 创建存储过程。选择数据库 carsales_info，单击 phpMyAdmin 页面中的“程序”超链接，在“新建”区域内单击“添加程序”按钮，进入程序编辑页面，输入程序名称 select_cars，选择程序类型为 PROCEDURE，设置参数（也可以删除参数设置创建无参过程），在文本编辑区输入文本，如图 18-48 所示。

② 生成存储过程。在图 18-48 中单击“执行”按钮，系统就创建一个存储过程 select_cars，如图 18-49 所示，如果有错误，系统会提示，可以根据提示进行修改。

③ 执行存储过程。单击图 18-49 中存储过程 select_cars 行的“执行”按钮，系统就会运行此存储过程，执行结果如图 18-50 所示。



图 18-48　创建存储过程

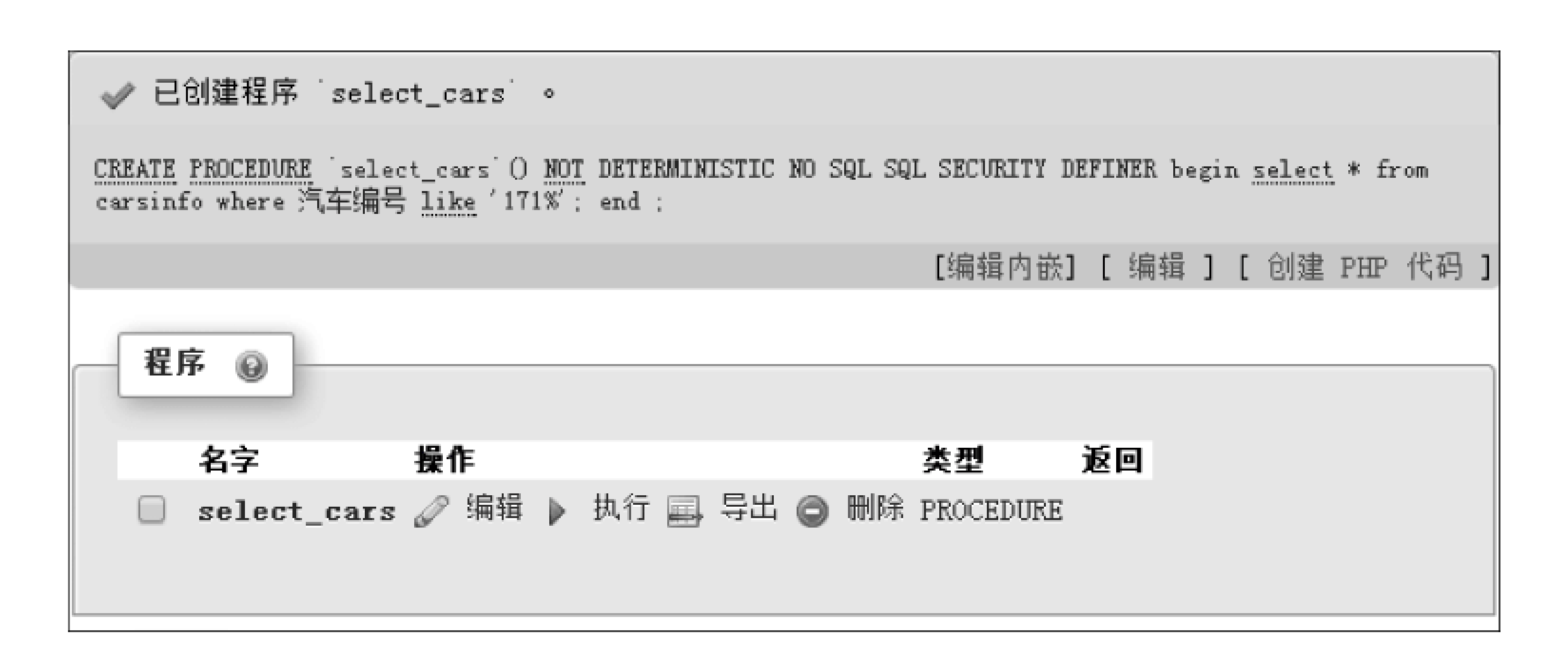


图 18-49　创建存储过程成功

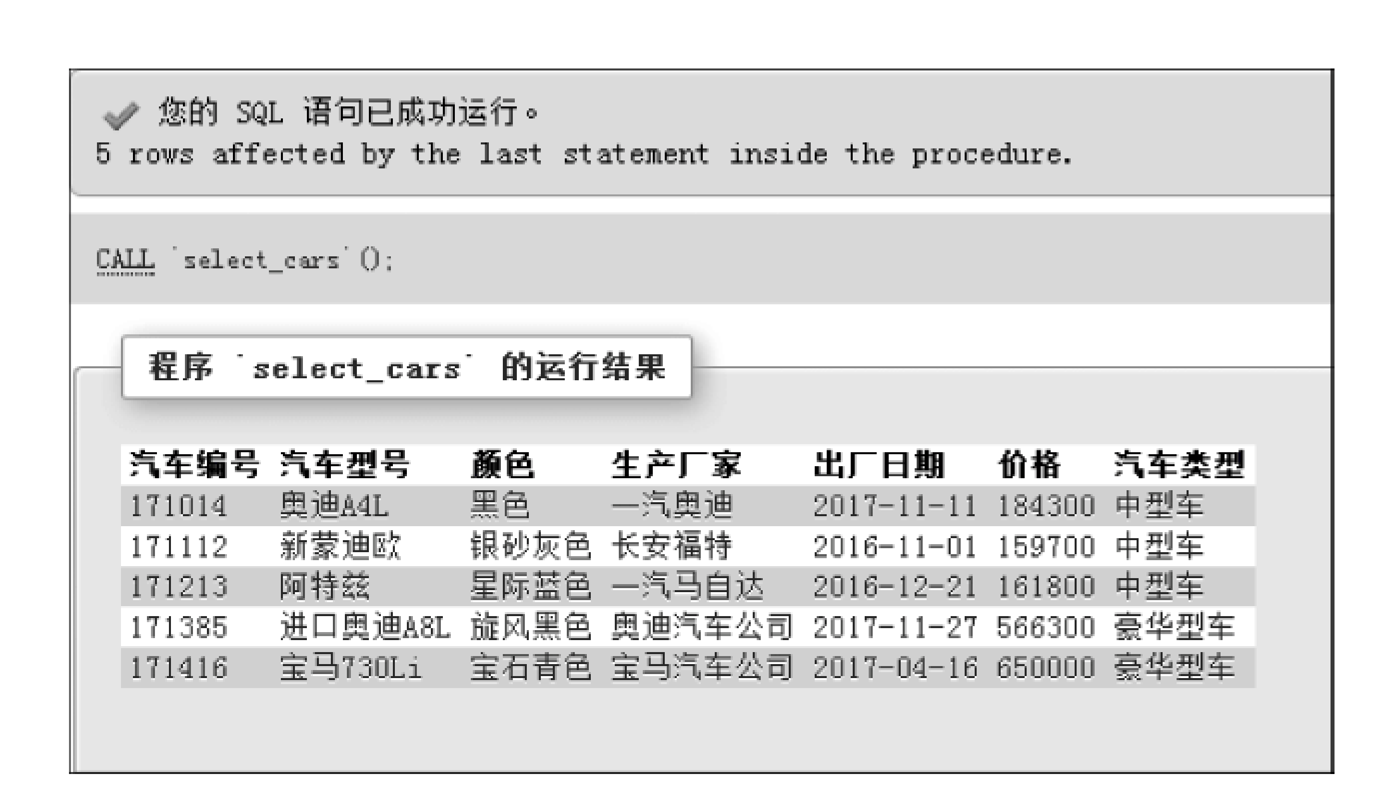


图 18-50　存储过程的运行结果

(8) 二进制日志的管理。在 phpMyAdmin 的主页面中，单击“二进制日志”超链接，可以查看当前 MySQL 数据库中的二进制日志。在“选择要查看的二进制日志”标签的下拉框中选择二进制日志文件，如图 18-51 所示。单击“执行”按钮，系统会显示此文件中的二进制事件表，如图 18-52 所示。如果选择“性能分析”，系统会自动显示系统详细的状态描述文件如图 18-53 所示。通过这些方法，可以对 MySQL 服务器的运行状态进行检测和观察，以进一步优化服务器的管理。

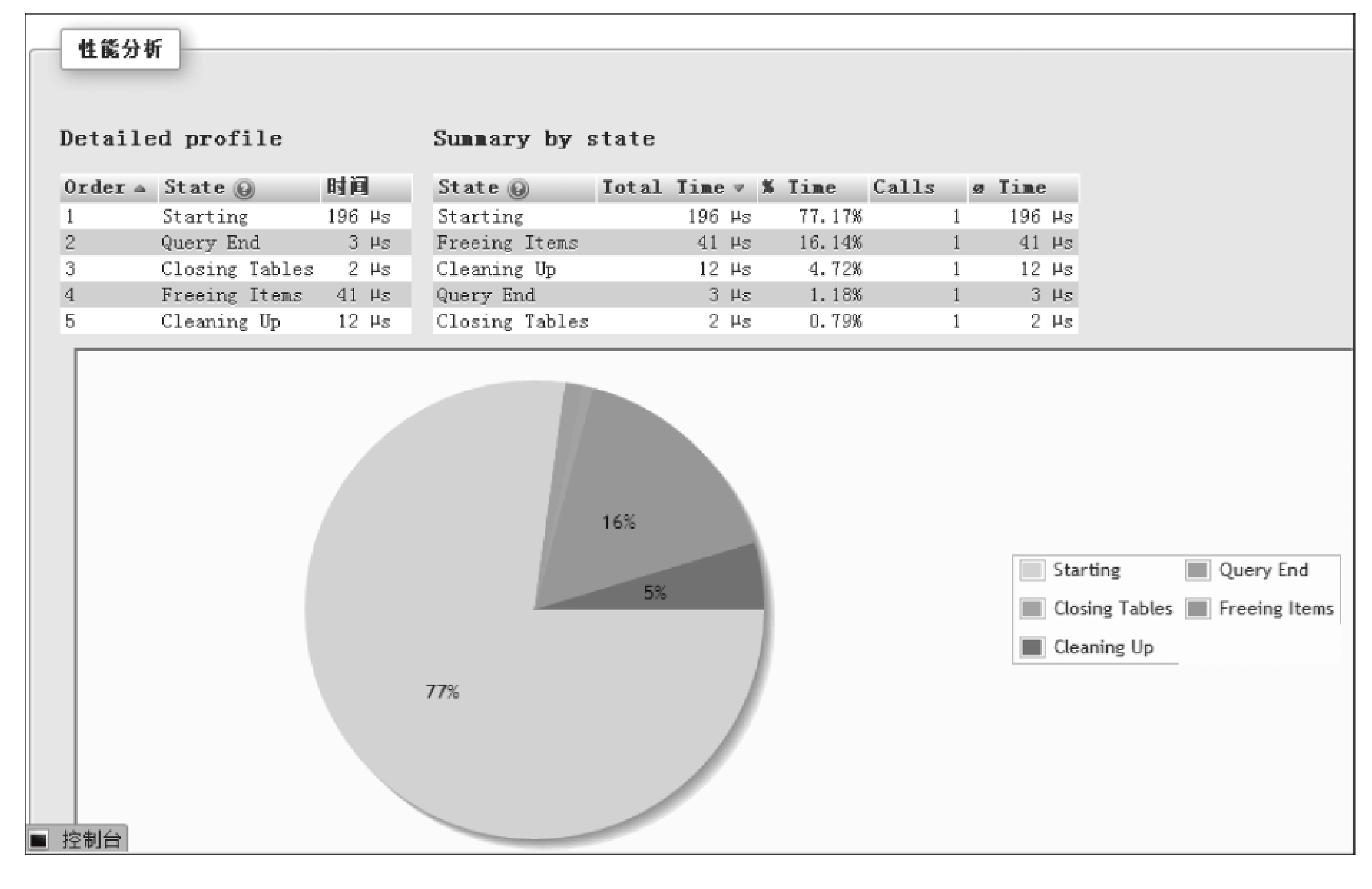


图 18-51　选择二进制日志文件



日志文件名	位置	事件类型	服务器ID	初始位置	信息
PGIG1MIWMYPOFBS-bin.000065	4	Format_desc	1	123	Server ver: 5.7.17-log, Binlog ver: 4
PGIG1MIWMYPOFBS-bin.000065	123	Previous_gtids	1	154	
PGIG1MIWMYPOFBS-bin.000065	154	Anonymous_Gtid	1	219	SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
PGIG1MIWMYPOFBS-bin.000065	219	Query	1	391	CREATE DATABASE `carsales_info` DEFAULT CHARSET=gb2312 COLLATE gb2312_chinese_ci

图 18-52　查看二进制日志文件内容



Order	State	时间
1	Starting	196 μs
2	Query End	3 μs
3	Closing Tables	2 μs
4	Freeing Items	41 μs
5	Cleaning Up	12 μs

State	Total Time	% Time	Calls	ø Time
Starting	196 μs	77.17%	1	196 μs
Freeing Items	41 μs	16.14%	1	41 μs
Cleaning Up	12 μs	4.72%	1	12 μs
Query End	3 μs	1.18%	1	3 μs
Closing Tables	2 μs	0.79%	1	2 μs

图 18-53　查看服务器性能指标

phpMyAdmin 软件功能强大，一般情况下，能够实现 MySQL 数据库管理的常用操作，在学习过程中，可以举一反三，尝试着去完成一些其他操作。同时，也可以采用可视化方法观察 MySQL 数据库服务器性能的指标参数，例如系统变量、字符集、插件、存储引擎、服务器运行状态等。还可以实现设置编码格式，创建 PHP 代码等操作。

4. 实验报告总结

（1）总结 phpMyAdmin 软件操作过程中常常出错的原因。

（2）总结 phpMyAdmin 软件与其他可视化软件相比，展现的些优势。

附：carsales_info 数据库的表结构和常用数据。

```
#数据库 - 汽车销售信息管理系统
 create database carsales_info;
 use carsales_info;
 #管理员表
 create table user(
   userId   char(10) primary key,
   username char(20),
   password char(10)
 );
 insert into user values('111111', 'admin', 'admin');
 #汽车信息表
 create table carsinfo(
   汽车编号 char(10) primary key,
   汽车型号 char(10),
   颜色     char(10),
   生产厂家 char(30),
   出厂日期 date,
   价格     int,
     汽车类型 char(20)
   );
insert into carsinfo values( '170111', '标致', '蓝色', '东风汽车', '2017 - 01 - 01', 177800, '小型车');
insert into carsinfo values( '170212', '吉利熊猫', '白色', '吉利汽车', '2017 - 02 - 11', 29700, '微型车');
insert into carsinfo values( '170314', '奔奔', '白色', '长安轿车', '2017 - 02 - 11', 34400, '微型车');
insert into carsinfo values( '170415', '大众 POLO', '绿色', '上海大众', '2017 - 03 - 11', 53900, '小型车');
insert into carsinfo values( '170516', '新威驰', '蓝色', '一汽丰田', '2016 - 10 - 01', 49800, '小型车');
insert into carsinfo values( '1750623', 'MG3', '红色', 'MG', '2017 - 11 - 01', 45700, '小型车');
insert into carsinfo values( '170715', '科鲁兹三厢', '白色', '上汽通用雪佛兰','2017 - 08 - 21', 73900, '紧凑型车');
insert into carsinfo values( '170815', '奥迪 A3', '红色', '一汽奥迪', '2017 - 09 - 09', 130000, '紧凑型车');
insert into carsinfo values( '170915', '大众高尔夫', '白色', '一汽大众','2017 - 10 - 11', 89400, '紧凑型车');
insert into carsinfo values( '171014', '奥迪 A4L', '黑色', '一汽奥迪', '2017 - 11 - 11', 184300, '中型车');
insert into carsinfo values( '171112', '新蒙迪欧', '银砂灰色', '长安福特', '2016 - 11 - 01',
```

```
159700, '中型车');
insert into carsinfo values( '171213', '阿特兹', '星际蓝色', '一汽马自达', '2016-12-21', 161800, '中型车');
insert into carsinfo values( '171385', '进口奥迪 A8L', '旋风黑色', '奥迪汽车公司', '2017-11-27', 566300, '豪华型车');
insert into carsinfo values( '171416', '宝马 730Li', '宝石青色', '宝马汽车公司', '2017-04-16', 650000, '豪华型车');
    #职工表
    create table employee(
      员工编号 char(20) primary key,
      姓名     char(10),
      性别     char(4),
      年龄     int,
      籍贯     char(30),
      学历     char(10)
   );
insert into employee values( 'e201701','余凯富','男',22,'山东青岛','本科');
insert into employee values( 'e201702','任盈盈','女',21,'北京朝阳','大专');
insert into employee values( 'e201703','岳灵珊','女',21,'上海虹口','硕士');
insert into employee values( 'e201704','林丽芬','男',23,'河北廊坊','博士');
insert into employee values( 'e201705','岳秋萍','男',30,'北京海淀','博士');
insert into employee values( 'e201706','仪项琳','女',28,'西藏拉萨','本科');
insert into employee values( 'e201707','林南理','男',32,'山西运城','硕士');
insert into employee values( 'e201708','左宗海','男',35,'湖南长沙','本科');
insert into employee values( 'e201709','向四伏','男',33,'海南三亚','专科');
insert into employee values( 'e201710','田回光','男',40,'浙江杭州','高中');
insert into employee values( 'e201711', '王子虚','男',36,'山西大同','初中');
insert into employee values( 'e201712','段开设','男',24,'陕西西安','本科');
insert into employee values( 'e201713','计非竹','男',23,'四川成都','小学');
   #客户表
   create table customer(
      客户编号 char(10) primary key,
      客户名称 char(10),
      性别     char(4),
      联系方式 char(12),
      地址     char(30)
   );
insert into customer values('c201701','李洪发','男','18712345678','蜀山');
insert into customer values('c201702','赵灵王','女','18723456781','苗疆');
insert into customer values('c201703','林月野','女','18734567812','林堡');
insert into customer values( 'c201704','林阿风','女','18745678123','南寨');
insert into customer values('c201705','景天印','男','18756781234','渝州');
insert into customer values('c201706','唐雪因','女','18767812345','渝达');
insert into customer values('c201707','龙先葵','女','18778123456','台湾');
insert into customer values('c201708','徐长卿','男','18781234567','巴蜀');
insert into customer values( 'c201709','南宫梅','男','18713456782','洞庭');
insert into customer values('c201710','姜云成','男','18714567823','欧河');
insert into customer values('c201711','唐雨白','女','18715678234','青镇');
   #业务表
   create table  business(
      员工编号 char(10),
```

```
        客户编号 char(10),
        联系时间 datetime,
        联系方式 char(20),
        联系结果 char(50)
    );
    #汽车销售信息表
    create table cars_sales(
        订单编号 char(20) primary key,
        汽车编号 char(10),
        销售日期 date,
        汽车类型 char(20),
        颜色     char(20),
        数量     int,
        销售价格 int,
        员工编号 char(10),
        客户编号 char(10)
    );
insert into cars_sales values( 'sale001', '10101', '2017-09-23', 'SUV', '黑色', 10, 120000, '
007', '110' );
```

第3部分
MySQL数据库模拟试题及参考答案

学习导读：

本部分包括6套MySQL数据库的模拟试题和参考答案，涵盖了本课程的主要知识点，可以帮助读者了解和检验自己的学习情况。

前4套以MySQL基本知识和基本操作为重点内容，第5套有个别题目是PHP软件内容，第6套添加了数据库设计方面的内容。

MySQL 数据库模拟试题 1

说明：本试题为笔试方式，考试时间 100 分钟，总计 100 分。

一、选择题（每题 1 分，共 20 分）

1. 下面标识符不合法的是________。

   A. null_1　　B. _mysql　　C. 3money　　D. trigger1

2. MySQL 数据库无处不在的数据危险不包括________。

   A. 系统故障　　B. 计划故障　　C. 介质故障　　D. 事务故障

3. 使用 select 将表中数据导出到文件，可以使用________子句。

   A. to file　　B. into file　　C. out to file　　D. into outfile

4. 一个关系只有一个________。

   A. 候选码　　B. 主码　　C. 外码　　D. 超码

5. 基于“学生-选课-课程”数据库中的 3 个关系：

```
stu(sno,sname,sex,department),主码是 sno
course(cno,cname,teacher),主码是 cno
sc(Sno,cno,grade),   主码是(sno,cno)
```

在下列关于保持数据库完整性的叙述中，________是不正确的。

   A. 向关系 sc 插入元组时，sno 和 cno 都不能是空值(null)

   B. 可以任意删除关系 sc 中的元组

   C. 向任何一个关系插入元组时，必须保证该关系主码值的唯一性

   D. 可以任意删除关系 cno 中的元组

6. 要想移除账户，应使用________语句。

   A. delete user　　B. drop user　　C. delete priv　　D. drop priv

7. 触发器可以创建在________中。

   A. 表　　B. 索引　　C. 事件　　D. 函数

8. 使用________语句删除触发器 trig_test。

   A. drop * from trig_Test

   B. drop trig_Tes

   C. drop triger where name='trig_Test'

   D. drop triger trig_Test

9. 在 SQL 语言中，子查询是________。

   A. 返回单表中数据子集的查询语句

   B. 选取多表中字段子集的查询语句

   C. 选取单表中字段子集的查询语句

   D. 嵌入到另一个查询语句之中的查询语句

10. 在数据库设计中，E-R 模型是进行________的一个主要工具。

A. 需求分析　　B. 概念设计　　C. 逻辑设计　　D. 物理设计

11. 通过以下________语句临时关闭事件 e_test。

A. alter event e_test disable　　B. alter event e_test drop
C. alter event e_test enable　　D. alter event e_test delete

12. 视图是一个虚表，视图的构造基于________。

A. 基表或视图　　B. 基表或索引
C. 视图或索引　　D. 基表、视图或索引

13. 设 s_avg(sno,avg_grade)是一个基于关系 sc 定义的学号和他的平均成绩的视图。下面对该视图的操作语句中，________是不能正确执行的。

Ⅰ. update s_avg set avg_grade=90 where sno=2017010601
Ⅱ. select sno, avg_grade from s_avg where sno=2017010601

A. 仅Ⅰ　　B. 仅Ⅱ　　C. 都能　　D. 都不能

14. 如果关系模式 R 属于 1NF，且每个非主属性都完全函数依赖于 R 的主码，则 R 属于________。

A. 2NF　　B. 3NF　　C. BCNF　　D. 4NF

15. 可用于从表或视图中检索数据的 MySQL 语句是________。

A. select　　B. insert　　C. update　　D. delete

16. MySQL 安装成功后，在系统中会默认建立一个________用户。

A. root　　B. user　　C. mysql　　D. sys

17. 当使用 select 语句返回的结果集中行数很多时，为了便于用户对结果数据的浏览和操作，可以使用________子句来限制被 select 语句返回的行数。

A. limit 语句　　B. insert 语句　　C. update 语句　　D. delete 语句

18. delete from Stu where 年龄>60 语句的功能是________。

A. 从 Stu 表中彻底删除记录
B. Stu 表中年龄大于 60 岁的记录被删除
C. 删除 Stu 表
D. 删除 Stu 表的年龄列

19. 索引可以提高哪一操作的效率? ________

A. insert　　B. update　　C. delete　　D. select

20. 在 MySQL 中，可以使用________语句删除表中的全部数据而保留表结构。

A. limit　　B. outer　　C. drop　　D. truncate

二、填空题(每空 1 分，共 10 分)

1. 在 E-R 图中，矩形表示________，椭圆形表示________。
2. 保存在 MySQL 安装目录中的选项文件名是________。
3. MySQL 安装包含典型安装、定制安装和________ 3 种安装类型。
4. 如果一个列是________类型，则可以使用________关键字指定该列的值自动增加。
5. 在 MySQL 中，通常使用________值来表示一个列是未确定值的情形。
6. 将一个列设置为主键的方法是，在列的定义中使用________子句。
7. 创建普通索引时，通常使用的关键字是________。

8. 如果使用________关键词,则即使在更新过程中出现错误,更新语句也不会中断。

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
delimiter //
create  function  count_pro (sid  int)  returns int
begin
return  (select count( * ) from arts where s_id = sid);
end  //
delimiter ;
```

2.
```
explain select * from  fruits where f_name like '%x';
explain select * from  fruits where f_name like 'x%';
```

3.
```
mysqldump  -u root  -p  --databases  booksDB test >
d:\backup\books_testDB_20170301.sql
```

4.
```
declare bookname char(50);
declare bookprice decimal(8,2);
select b_name,b_price into bookname, bookprice
from  books where b_id = 'a1101';
```

5.
```
create index index_name on fruits(f_name);
explain select * from  fruits where f_name = 'apple';
```

6.
```
declare id int default 0;
add_loop: loop
set id = id + 1;
if id >= 10 then  leave add_loop;
end if;
end loop add_ loop;
```

四、简答题(每小题 3 分,共 12 分)

1. 简述性能优化的目标和功能。
2. 简述共享锁与排他锁的作用。
3. 简述用户访问 MySQL 服务器数据的工作流程。
4. 简述使用视图的优点。

五、综合题(每小题 4 分,共 40 分)

现有 MySQL 数据库 teach 中的 3 个表,表结构分别描述如下:

```
student(学号 char(9) primary key,姓名 char(10),住址 varchar(20),性别 char(2),系名 char(12))
course(课程号 char(6) primary key,课程名 char(12),学时 int,教师编号 char(6))
c_select (学号 char(9),课程号 char(6),成绩 int )
```

按要求使用 MySQL 语句完成操作。

1. 利用 Alter Table 命令在 c_select 表中添加主键“学号+课程号”。并创建外键 fk_1,参照 student 表的学号。

2. 查询姓名为“黄岩”的学生所选课程的“课程名”及“成绩”。

3. 创建视图 V_update,检索所有选修课程号为 c05126 的学生的学号、课程号及成绩。

4. 创建存储过程 upcourse 修改课程名称,提示输入参数为课程号,输出参数为旧的课程名称与新的课程名称。

5. 调用存储过程,并查看课程名的修改情况。

6. 添加登录名为 tuser，密码为 123abc 的 MySQL 登录账户。用户 tuser 对当前数据库中的所有表有查询、插入权限，并授予 grant 权限。

7. 使用 revoke 语句收回 tuser 用户的所有权限，包括 grant 权限。

8. 使用 mysqldump 命令备份数据库中的 c_select 表，备份路径为 d:/bak/，备份文件名为 teach_cs.sql。

9. 创建一个 after 触发器，要求实现以下功能：在 c_select 表上创建一个触发器 tr_select，当在 c_select 表字段中修改“成绩”后，触发该触发器，检查“成绩”值是否在 0～100。

10. 使用 select into...outfile 语句，备份表 student 的全部数据到 d 盘的 bak 目录下一个名为 backupstudent.txt 的文件中，要求字段值如果是字符则用双引号标注，字段值之间用逗号隔开，每行以问号为结束标志。

MySQL 数据库模拟试题 1 参考答案

一、选择题(每题 1 分,共 20 分)

1～5. CBDBD　6～10. BADDB　11～15. AAAAA　16～20. AABDA

二、填空题(每空 1 分,共 10 分)

1. 实体　属性
2. my.ini
3. 完全安装
4. 整数,“auto increment”
5. null
6. primary key
7. index
8. ignore

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1. 创建存储过程,输出 arts 表的指定记录的数量。
2. 利用 explain 分析比较查询语句中使用 like 关键字,并且匹配的字符串中含有“%”符时的结果。
3. 使用 mysqldump 备份 booksDB 和 test 数据库。
4. 声明变量 bookname 和 bookprice,通过 select … into 语句查询指定记录并为变量赋值。
5. 创建索引,分析使用索引时的查询情况。
6. 使用 loop 语句进行循环操作,id 值小于等于 10 之前,将重复执行循环过程。

四、简答题(每小题 3 分,共 12 分)

1. 优化性能是通过某些高效的方法提高 MySQL 数据库的性能,其目的是为了使 MySQL 数据库运行速度更快、占用的磁盘空间更小。性能优化包括优化查询速度、优化更新速度、优化 MySQL 服务器等。优化 MySQL 数据库是数据库管理员的必备技能。

在实际工作中,数据的查询优化可以有效地提高 MySQL 数据库的性能。一个成功的数据库应用系统的开发,在查询优化方面一定会付出资源。对查询优化的处理,不仅会影响到数据库的工作效率,而且会给社会带来较高的效益。

2. 排他锁又称为 X 锁。如果事务 T1 获得了数据行 D 上的排他锁,则 T1 对数据行既可读又可写。事务 T1 对数据行 D 加上排他锁,则其他事务对数据行 D 的任务封锁请求都不会成功,直至事务 T1 释放数据行 D 上的排他锁。

共享锁又称为 S 锁。如果事务 T1 获得了数据行 D 上的共享锁,则 T1 对数据项 D 可以读但不可以写。事务 T1 对数据行 D 加上共享锁,则其他事务对数据行 D 的排他锁请求不会成功,而对数据行 D 的共享锁请求可以成功。

3. 用户访问 MySQL 服务器数据的工作流程具体包括如下内容:

(1) 数据库用户访问 MySQL 服务器的数据时，首先需要选择一台登录主机，然后在该登录主机上开启客户机，输入正确的账户名、密码，建立一条客户机与服务器之间的“通信链路”。

(2) 接着数据库用户就可以在 MySQL 客户机上输入 MySQL 命令，这些命令沿着该通信链路传送给 MySQL 服务实例，这个过程称为客户机向 MySQL 服务器发送请求。

(3) 服务实例负责解析这些 MySQL 命令，并选择一种执行计划运行这些命令，然后将执行结果沿着通信链路返回给客户机，这个过程称为服务器向 MySQL 客户机返回响应。

(4) 数据库用户关闭客户机，通信链路被断开，该客户机对应的会话结束。

4. 视图为数据库用户提供了很多的便利，主要包括以下几个方面：

(1) 简化数据查询和处理。视图可以为用户集中多个表中的数据，简化用户对数据的查询和处理。

(2) 屏蔽数据库的复杂性。数据库表的更改不影响用户对数据库的使用，用户也不必了解复杂的数据库中的表结构。例如，那些定义了若干张表连接的视图，就将表与表之间的连接操作对用户隐蔽起来了。

(3) 安全性。如果想要使用户只能查询或修改用户有权限访问的数据，也可以只授予用户访问视图的权限，而不授予访问表的权限，这样就提高了数据库的安全性。

五、综合题（每小题 4 分，共 40 分）

1.
```
alter  table c_select
add primer key (学号,课程号);
alter table c_select add constraint fk_1 foreign key (学号)
references student(学号)
    on update restrict
    on delete set null;
```

2.
```
select s.姓名,k.课程名,x.成绩
from course k,student s, c_select x
where k.课程号 = x.课程号 and s.学号 = x.学号 and s.姓名 = '黄岩';
```

3.
```
create view V_update
as
select * from c_select where 课程号 = 'c05126';
select * from v_update;
```

4.
```
delimiter //
create procedure upcourse(in courseid char(6),in newname1 char(9),
out oldname char(9), out newname2 char(9))
begin
  set newname2 = newname1;
  select oldname = 课程名 from course
                      where 课程号 = courseid;
  update course set 课程名 = newname1
                     where 课程号 = courseid;
end //
delimiter;
```

5.
```
select * from course;
exec upcourse ('c05126','MySQL', @name1,@name2);
select @name1 + '已更改为: ' + @name2;
```

6. grant select,insert on * to 'tuser'@'localhost'
   identified by '123abc '
   with grant option;

7. revoke all privileges,grant option
   from tuser@localhost;

8. mysqldump -u root -p teach c_select > d:/bak/teaching_ss.sql

9. delimiter //
   create triger tr_select on c_select
   after update for each row
   begin
   update from c_select set c_select.成绩 = old.成绩
   where new.成绩> 100 or new.成绩< 0;
   end//
   delimiter ;

10. select * from student
    into outfile 'd:/bak/backupstudent.txt'
    fields
      terminated by','
      optionally enclosed by ""
    lines
       terminated by'?';

MySQL 数据库模拟试题 2

说明：本试题为笔试方式，考试时间 100 分钟，总计 100 分。

一、选择题(每题 1 分，共 20 分)

1. 语句“select dayofmonth(curdate())　'month';”的输出结果为________。

A. month　　B. 当前日期　　C. 本月第几天　　D. 当前月份

2. 候选码的属性可以有________。

A. 多个　　B. 0 个　　C. 1 个　　D. 1 个或多个

3. 在 MySQL 的安装过程中，若选用“启用 TCP/IP 网络”，则 MySQL 会默认选用的端口号是________。

A. 1521　　B. 5560　　C. 3306　　D. 80

4. 一个触发器能定义在________表中。

A. 一个　　B. 一个或者多个　　C. 一个到三个　　D. 任意多个

5. 在 MySQL 中，通常使用________语句来进行数据的检索、输出操作。

A. select　　B. insert　　C. delete　　D. update

6. 在 select 语句中，可以使用________子句，将结果集中的数据行根据选择列的值进行逻辑分组，以便能汇总表内容的子集，即实现对每个组的聚集计算。

A. limit　　B. group by　　C. where　　D. order by

7. 查看现在的数据库 mysqltest 中所有表的命令是________。

A. show databases;　　B. create database mysqltest;

C. use mysqltest;(回车)　　D. show tables;

8. 用文本方式将数据装入数据库表中的命令是________。

A. load data infile “D:/mysql. txt” into table mytable;

B. source d:/mysql. sql;

C. drop table mytable;

D. delete from mytable;

9. 用于显示表的结构的命令是________。

A. create table mytable (name varchar(20), sex char(1));

B. describe mytable;

C. insert into mytable values ("hyq","M");

D. update mytable set sex="f" where name='hyq';

10. 在 mysql >提示行下输入________可以显示帮助命令。

A. help　　B. helps　　C. \h　　D. A 或 C

11. 在 MySQL 中，可以使用 insert 或________语句，向数据库中一个已有的表插入一行或多行元组数据。

A. select　　B. insert　　C. replace　　D. update

12. 在 MySQL 5.7 以上版本的系统中，默认的存储引擎是________。

A. MyISAM　　B. MEMORY　　C. InnoDB　　D. ARCHIVE

13. MySQL 系统中，表结构文件的扩展名是________。

A. .frm　　B. .myd　　C. .myi　　D. .mdf

14. ________备份是在某一次完全备份的基础上，只备份其后数据的变化。

A. 比较　　B. 检查　　C. 增量　　D. 二次

15. 设属性 A 是关系 R 的主属性，则属性 A 不能取空值(null)，这是________。

A. 参照完整性规则　　B. 用户定义完整性规则

C. 关系完整性规则　　D. 实体完整

16. 设关系 R 和 S 的元组个数分别为 100 和 300，关系 T 是 R 与 S 的笛卡儿积，则 T 的元组个数为________。

A. 400　　B. 10000　　C. 30000　　D. 90000

17. 创建表时，不允许某列为空可以使用________。

A. not null　　B. no null　　C. not blank　　D. no blank

18. 下列________类型不是 MySQL 中常用的数据类型。

A. int　　B. var　　C. time　　D. char

19. 不是外键定义中 on delete 与 on update 的设置是________。

A. cascade　　B. action　　C. restrict　　D. set null

20. select * from city　limit　3,7 描述正确的是________。

A. 获取第 3 条到第 7 条记录　　B. 获取第 3 条到第 10 条记录

C. 获取第 4 条到第 10 条记录　　D. 获取第 1 条到第 21 条记录

二、填空题(每空 1 分，共 10 分)

1. MySQL 数据库所支持的 SQL 语言主要包含________、________、________和 MySQL 扩展增加的语言要素 4 个部分。

2. 创建唯一性索引时，通常使用的关键字是________。

3. 在创建表格时，可以使用________关键字使当前建立的表为临时表。

4. 在 MySQL 中，可以使用________语句来实现权限的撤销。

5. E-R 图是数据库设计的工具之一，它一般适用于建立数据库的________。

6. ________关键字能够将两个或多个 select 语句的结果连接起来。

7. ________是表、视图、存储过程、触发器等数据库对象的集合，是数据库管理系统的核心内容。

8. select 语句中，若希望查询的结果不出现重复元组，则应在 select 子句中使用________保留字。

三、阅读程序，指出下列程序段的功能(每题 3 分，共 18 分)

1.
```
select 60 + round(rand() * 40,0),  60 + round(rand() * 40,0);
```

2.
```
create view v_tab1
as
select quantity, price, quantity * price
from  tab1;
select * from  v_tab1;
```

3.
```
delimiter //
create  procedure pro_score()
  begin
    select * from  score;
  end //
delimiter ;
```

4.
```
select  sname  from   student
      where sno not  in
            ( select s_no  from  score  where  final < 60);
```

5.
```
grant select, update (id, name, age)  on  testdb.student
To  'Admin1'@'localhost'  identified by  'pword'
with max_connections_per_hour 30;
```

6.
```
mysqldump  -t  d:\backup testDB student  -u root  -p
 --fields-terminated-by=,
 --fields-optionally-enclosed-by=\"
 --fields-escaped-by=?
 --lines-terminated-by=\r\n
```

四、简答题(每题 3 分,共 12 分)

1. 简述连接查询的分类和特点。
2. 在 MySQL 中可以授予的权限有哪几组?
3. 简述事件与触发器的区别。
4. 简述数据完整性的分类和特点。

五、综合题(每小题 4 分,共 40 分)

使用 MySQL 命令完成操作。

1. 创建一个新的数据库,名称为 student。

2. 在数据库 student 中,创建一个学生信息表 stu_info,其中 sname 字段具有唯一性。表结构如下:

```
stu_info(sno char(10) ,sname char(10),sex char(2),
      birthday date,polity  char(10))
```

3. 向表 stu_info 中插入 1 条记录,数据如下:

学号: 193552017,姓名: 王秋桐,性别: 女,出生日期: 1999_09_09。

4. 查询 stu_info 表中新插入的记录信息。

5. 将 polity 的值为空的记录用'文学'代替 null 显示。

6. 查询 stu_info 表,将返回的记录数赋给变量@RowsReturn,并显示。

7. 创建一个视图 v_stu,要求显示 stu_info 表中所有姓黄的同学的所有信息。

8. 向表 stu_info 中添加一个“备注”字段,数据类型为 char(40)。“备注”字段允许为 null。

9. 按性别分组,求出 stu_infot 表中的男女学生的人数。

10. 将对数据库 teach 中 stu_info 表的 select 权限授予数据库用户 hans。

MySQL 数据库模拟试题 2 参考答案

一、选择题(每题 1 分,共 20 分)

1～5. DDCAA　6～10. BDABD　11～15. CCACA　16～20. CABBC

二、填空题(每空 1 分,共 10 分)

1. 数据定义语言(DDL)　数据操纵语言(DML)　数据控制语言(DCL)
2. unique
3. temporary
4. revoke
5. 概念模型
6. union
7. 数据库
8. distinct

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1. 使用数学函数 rand()生成 2 个 60～100 的随机整数。
2. 创建一个名为 v_tab1 的视图,并显示视图中的所有数据。
3. 创建一个查看 score 表的存储过程 pro_score(),每次调用这个存储过程的时候都会执行 select 语句查看表的数据。
4. 查询期末成绩全部及格的学生名单。
5. 创建新账户,用户名称为 Admin1,密码为 pword,允许其从本地主机访问 MySQL。对学生表有查询和修改权限,且每小时可以连接服务器不超过 30 次。
6. 使用 mysqldump 命令将 test1 数据库中的 person 表中的记录导出到文本文件,使用 fields 选项,要求字段之间使用逗号“,”间隔,所有字符类型字段值用双引号括起来,定义转义字符为问号“?”,每行记录以回车换行符“\r\n”结尾。

四、简答题(每题 3 分,共 12 分)

1. 连接运算符可以实现多个表查询。连接可分为三种:内连接、外连接和交叉连接。连接操作中的连接条件由被连接表中的列和比较运算符、逻辑运算符构成。无论哪种连接都不能对 text、ntext 和 image 数据类型列进行直接连接。

内连接(inner join)使用比较运算符进行表间某(些)列数据的比较操作,并列出这些表中与连接条件相匹配的数据行。

外连接分为左外连接、右外连接。与内连接不同的是,外连接不只列出与连接条件相匹配的行,而且列出左表、右表中所有符合搜索条件的数据行。

交叉连接(cross join)没有 where 子句,它返回连接表中所有数据行的笛卡儿积,其结果集合中的数据行数等于第一个表中符合查询条件的数据行数乘以第二个表中符合查询条件的数据行数。

2. 在 MySQL 中可以授予的权限有这样几组:列权限;表权限;数据库权限;用户权限。

3. 事件和触发器相似，都是在某些事情发生的时候才被启动，因此事件也可称作临时触发器(temporal trig-ger)。其中，事件是基于特定时间周期触发来执行某些任务，而触发器是基于某个表所产生的事件触发的，它们的区别也在于此。

4. 确保持久化数据检索不出错对于数据管理来说非常关键，数据完整性，能保证查询结果的正确性，数据完整性的分类和特点如下：

(1) 实体完整性。实体完整性是指关系的主关键字不能取"空值"。一个关系对应现实世界中的一个实体集。在关系模式中，以主关键字作为唯一性标识，而主关键字中的属性(称为主属性)不能取空值。

(2) 域完整性确保属性中只允许一个有效数据。域是属性可能值的范围，如整数、日期或字符。是否可以是空值也是域完整性的一部分。

(3) 参照完整性。参照完整性是定义建立关系之间联系的主关键字与外部关键字引用的约束条件。关系数据库中通常包含多个存在相互联系的关系，关系与关系之间的联系是通过公共属性来实现的。

(4) 事务完整性。事务可以确保每个逻辑单元的工作(如插入 100 行或更新 1000 行数据)作为单个事务执行。事务可通过其 4 个基本属性检测数据库产品的质量，即原子性、一致性、隔离性和持久性。

(5) 用户定义完整性。用户定义完整性则是根据应用环境的要求和实际的需要，对某一具体应用所涉及的数据提出约束性条件。用户定义完整性主要包括字段有效性约束和记录有效性。

五、综合题(每小题 4 分，共 40 分)

1.
```
create database student;
```

2.
```
use student;
create table stu_info
( sno char(10)  primary key,
     sname char(8) not null,
     sex char(2) not null ,
     birthday date  not null,
 polity  char(10) unll,
     constraint  uk_name  unique(sname)
);
```

3.
```
insert into stu_info (Sno,sname,sex)
values('193552017','王秋桐','女','1999_09_09');
```

4.
```
select sno as 学号, sname as 姓名,sex as 性别, polity as 社团
from  stu_info
where sno = '193552017';
```

5.
```
update stu_info  set polity =  '文学'
where  polity is null;
```

6.
```
select @RowsReturn = count( * )  from  stu_info;
#显示变量的值
select @RowsReturn  'select 返回的记录数';
```

7.
```
create view v_stu
as
select  *  from  stu_info
```

```
where  sname  like  '黄%';
```

8.
```
alter table stu_info  add  column  备注 char(40) null;
```

9.
```
select 单位 ,count( * ) as '人数'
from stu_infot
group by sex;
```

10.
```
grant select on teach. stu_info
to 'hans'@'localhost';
```

MySQL 数据库模拟试题 3

说明：本试题为笔试方式，考试时间 100 分钟，总计 100 分。

一、选择题（每题 1 分，共 20 分）

1. 在 MySQL 数据库中 null 表示________。

A. 0　　B. 空格　　C. 未确定值　　D. 其他

2. 语句块是一个 MySQL 语句集合，以________结束。

A. begin　　B. “}”　　C. go　　D. end

3. 存储程序中选择语句有________。

A. if　　B. while　　C. select　　D. switch

4. 语句 select * from city limit 5,10 描述正确的是________。

A. 获取第 6 条到第 10 条记录

B. 获取第 5 条到第 10 条记录

C. 获取第 6 条到第 15 条记录

D. 获取第 5 条到第 15 条记录

5. 视图是一个“虚表”，视图的构造基于________。

A. 基本表　　B. 视图　　C. 基本表或视图　　D. 数据字典

6. 使用 select 将表中数据导出到文件，可以使用________子句。

A. to file　　B. into file　　C. out to file　　D. into outfile

7. 下列________语句用来定义游标。

A. create　　B. declare

C. declare ...cursor for ...　　D. show

8. 下列说法中错误的是________。

A. 常用触发器有 insert、update 和 delete 3 种

B. 对于同一张数据表，可以同时有两个 before update 触发器

C. new 表在 insert 触发器中用来访问被插入的行

D. old 表中值只读不能被更新

9. 删除列的指令是________。

A. alter table ... delete ...　　B. alter table ... delete column...

C. alter table ... drop ...　　D. alter table ... drop column...

10. ________中提供了执行 mysqldump 之后对数据库的更改进行复制所需要的信息的功能。

A. 二进制日志文件　　B. MySQL 数据库

C. MySQL 配置文件　　D. bin 数据库

11. 不可对视图执行的操作有________。

A. select　　B. insert　　C. delete　　D. create index

12. 下列语句中，________不是表数据的基本操作语句。

A. create　　B. insert　　C. delete　　D. update

13. 下列描述正确的是________。

A. 一个数据库只能包含一个数据表　　B. 一个数据库可以包含多个数据表

C. 一个数据库只能包含两个数据表　　D. 一个数据表可以包含多个数据库

14. 设置表的默认字符集关键字是________。

A. default character　　B. default set

C. default　　D. default character set

15. 使用 insert 插入记录时，可使用________关键字忽略会导致重复关键字的错误记录。

A. no same　　B. ignore　　C. repeat　　D. unique

16. 对于 replace 语句，描述错误的是________。

A. replace 语句返回一个数字以表示受影响的行，包含删除行和插入行的总和

B. 通过返回值可以判断是否增加了新行还是替换了原有行

C. 因主键重复插入失败时直接更新原有行

D. 因主键重复插入失败时先删除原有行再插入新行

17. 要快速完全清空一张表可使用________语句。

A. truncate table　　B. delete table

C. drop table　　D. clear table

18. 使用 create view 创建视图时，如果给定了________子句，能替换已有视图。

A. all replace　　B. OR replace　　C. replace　　D. replace all

19. 通过以下哪个语句临时关闭事件 e_test？________

A. alter event e_test disable　　B. alter event e_test drop

C. alter event e_test enable　　D. alter event e_test delete

20. delete from S where 年龄> 60 语句的功能是________。

A. 从 S 表中彻底删除记录　　B. S 表中年龄大于 60 岁的记录被删除

C. 删除 S 表　　D. 删除 S 表的年龄列

二、填空题(每空 1 分，共 10 分)

1. 服务器是用来________、________、________和停止 MySQL 服务的。

2. 以@作为首部的变量在 MySQL 系统中称为________。

3. select 语句的完整语法较复杂，但至少包括的部分有________、________；使用关键字________可以把重复行屏蔽，将多个查询结果返回一个结果集合的运算符是________。

4. 声明游标的命令是________，关闭游标的命令是________。

5. MySQL 安装包含典型安装、定制安装和________三种安装类型。

三、阅读程序，指出下列程序段的功能(每题 3 分，共 18 分)

1.
```
select * from  course  left  join  score
on course.cno = score.cno;
```

2.
```
select date_sub('2017-01-02', interval 31 day) as col1,
    subdate('2017-01-02', interval 31 day) as col2;
```

3.
```
delimiter //
 create function fname ()
 returns char(20)
 begin
 return   (select sname  from  student  where phone =  '78548075');
 end  //
 delimiter ;
```

4.
```
explain select * from  score  where final > 90;
```

5.
```
grant  usage  on  *.*  to'testUser'@'localhost' identified by  ' pwdone ';
```

6.
```
select * from  test.student  into outfile "D:/student.txt"
fields
    terminated by ','
    enclosed by '\"'
    escaped by '\''
lines
    terminated by '\r\n';
```

四、简答题(每题 3 分,共 12 分)

1. 简述事件与触发器的区别。
2. 简要说明事务的原子性的特点。
3. 列举 MySQL 的系统特性。
4. MySQL 数据库备份与恢复的常用方法有哪些?

五、综合题(每小题 4 分,共 40 分)

使用 MySQL 语句完成操作。现有一个关于设备管理系统的数据库 equip_gl,数据库中有下列 3 个表,参考表结构如下:

(1) 设备表 equipment(eno,ename,ebuyday,etypeid,etypename);

其属性含义依次为设备编号(主键),设备名称,购入日期,设备分类号,设备分类名。

(2) 雇员表 employee(pno,pname,psex,pbirth,pposition);

其属性依次为员工号(主键),姓名,性别,出生日期,岗位。

(3) 设备使用表 u_equip(pno, eno ,borrowday,useday,ufee);

其属性依次为设备编号,员工号,借出日期,使用天数,收费金额。

1. 创建使用表 u_equip。(要求:包含主键约束和 check 约束检查“收费金额>=50”)。
2. 添加一个人员记录,员工号为 17009,姓名为“韩伟和”,性别为“男”,出生日期为 1993-10-08,职位为“副主任”。
3. 将设备名称为“复印机”的收费金额修改为 360。
4. 删除姓名为“马怡闻”的人员信息。
5. 查询每种设备平均收费金额超过 500 元的设备编号和平均收费金额。
6. 检索借用设备的收费金额超过 2000 元的员工姓名、设备编号、设备名称。
7. 检索借出设备借用 90 天以上的员工姓名、设备编号。
8. 将设备使用 u_equip 的查询、修改权限授予用户 manager,并允许该用户将此权限授予其他用户。
9. 定义视图 v_employee,完成如下功能:输出没有使用设备的人员信息。
10. 创建一个 after 触发器,要求实现以下功能:在 pmployee 表上创建一个触发器 tr_pmployee,修改 pmployee 表的字段 pno 数据后,触发该触发器,同时 u_equip 的 pno 数据更新。

MySQL 数据库模拟试题 3 参考答案

一、选择题(每题 1 分,共 20 分)

1～5. CDACC　6～10. DCBDA　11～15. BABDB　16～20. CABAB

二、填空题(每空 1 分,共 10 分)

1. 启动　暂停　重启
2. 会话变量
3. select　from　distinct
4. declare　close
5. 完全安装

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1. 以课程号为连接条件,利用左连接查询所有课程的成绩。
2. 使用 date_sub 和 subdate 函数执行日期减法操作。
3. 创建一个存储函数,参数定义为空,返回一个 char(20)类型的结果。
4. 分析查询 select * from　score where final＞90。
5. 使用 grant 语句将 testUser 用户的密码修改为“pwdone”。
6. 使用 select...into outfile 将 test 数据库中的 student 表中的记录导出到文本文件,使用 fields 选项和 lines 选项,要求字段之间使用逗号“,”间隔,所有字段值用双引号括起来,定义转义字符为单引号“\”。

四、简答题(每题 3 分,共 12 分)

1. 事件和触发器相似,都是在某些事情发生的时候才被启动,因此事件也可称作临时触发器(temporal trig-ger)。其中,事件是基于特定时间周期触发来执行某些任务,而触发器是基于某个表所产生的事件触发的,它们的区别也在于此。

2. 事务的所有操作必须完成,否则事务将被撤销。也就是说,事务是作为一个整体的单位处理的,不可以被分割。

3. MySQL 数据库的应用非常广泛,主要是因为其具有 4 个方面的优势。

(1) MySQL 数据库是开放源代码的数据库。任何人都可以获取 MySQL 数据库的源代码,可以修改 MySQL 数据库的缺陷,并以任何目的来使用该数据库。

(2) MySQL 数据库的跨平台性。MySQL 不仅可以在 Windows 系列的操作系统上运行,还可以在 UNIX、Linux 和 Mac OS 等操作系统上运行。

(3) MySQL 的价格优势。MySQL 数据库是一款自由软件,社区版本的 MySQL 数据库软件都是免费使用的,相对于 Oracle、SQL Server 和 DB2 这些价格昂贵的商业软件,MySQL 数据库具有绝对的价格优势。

(4) 功能强大、使用方便。MySQL 数据库是一个多用户、多线程的 SQL 数据库服务器,它是 C/S 结构的实现,由一个服务器守护程序 mysqlld 和很多不同的客户程序及库组成。

4. MySQL 数据库备份与恢复的常用方法有：

(1) 使用 SQL 语句备份和恢复表数据。

(2) 使用 MySQL 客户端实用程序备份和恢复数据。

(3) 使用 MySQL 图形界面工具备份和恢复数据。

(4) 直接利用复制完成备份和恢复。

五、综合题(每小题 4 分，共 40 分)

1.
```
create table u_equip(
pno char(8) ,
eno,char(8),
borrowday date,
useday   samllint ,
fee samllint constrint   check(fee> = 50),
primary key(pno, eno)
);
```

2.
```
insert into pmployee(pno,pname,psex,pbirth,pposition)
values('17009', '韩伟和', '男', '1993-10-08', '副主任');
```

3.
```
update   u_equip    set fee = 360    where eno in (
select eno from   equipment
where ename = '复印机');
```

4.
```
delete from employee
where pname = '马怡闻';
```

5.
```
select eno,avg(fee)   from u_equip
group by eno   having   avg(fee)> 500;
```

6.
```
select pname, equipment.eno,ename
from   equipment, u_equip, employee
where equipment.eno =  u_equip.eno and employee.pno =  u_equip.pno and fee > 2000
order by pname;
```

7.
```
select pname, employee.eno
from   employee, u_equip
where and employee.pno =  u_equip.pno and useday> = 90
```

8.
```
grant select,update on u_equip
to manager@localhost   with grant option;
```

9.
```
create view v_employee
as
select pno,pname,psex,pbirth,pposition
from pmployee
where pno not in ( select   pno   from u_equip);
```

10.
```
delimiter //
create triger tr_pmployee on pmployee
after update for each row
begin
  update u_equip set   pmployee.pno = new.pno
  where pmployee.pno = old.pno;
end//
delimiter;
```

MySQL 数据库模拟试题 4

说明：本试题为笔试方式，考试时间 100 分钟，总计 100 分。

一、选择题（每题 1 分，共 20 分）

1. 根据关系模式的完整性规则，一个关系中的主键________。

A. 不能由两列组成　　B. 不能成为另一个关系的外部键

C. 不允许空值　　D. 可以取空值

2. MySQL 使用________文件中的配置参数。

A. mylarger.ini　　B. mysmall.ini　　C. default.ini　　D. my.ini

3. 下列关于统计函数 count(字符串表达式)的叙述中，正确的是________。

A. 返回字符表达式中值的个数，即统计记录的个数

B. 统计字段应该是数字数据类型

C. 字符串表达式中可以是字段名

D. 以上都不正确

4. 下面________不是备份数据库的理由。

A. 数据库崩溃时恢复

B. 将数据从一个服务器转移到另外一个服务器

C. 记录数据的历史档案

D. 转换数据

5. 以下哪个语句用于撤销权限？________

A. delete　　B. drop　　C. revoke　　D. update

6. 选择要执行操作的数据库，应该用________ MySQL 命令。

A. use　　B. go　　C. exec　　D. DB

7. 下面关于数据库中表的行和列的叙述正确的是________。

A. 表中的行是有序的，列是无序的　　B. 表中的列是有序的，行是无序的

C. 表中的行和列都是有序的　　D. 表中的行和列都是无序的

8. 在 MySQL 语言中，条件“age between 20 and 30”表示年龄为 20～30，且________。

A. 包括 20 岁和 30 岁　　B. 不包括 20 岁和 30 岁

C. 包括 20 岁但不包括 30 岁　　D. 包括 30 岁但不包括 20 岁

9. 显示当前所有数据库的命令是________。

A. show databases　　B. show database

C. list databases　　D. list database

10. 有订单表 orders，包含用户信息 userid，产品信息 productid，以下________语句能够返回至少被订购过两回的 productid。

A. select productid from orders where count(productid)>1

B. select productid from orders where max(productid)>1

C. select productid from orders where having count(productid)>1 group by productid

D. select productid from orders group by productid having count(productid)>1

11. 在实际使用中,MySQL 所支持的触发器有 3 种,不包括________触发器。

A. insert　　B. delete　　C. update　　D. select

12. 只修改列的数据类型的指令是________。

A. alter table … alter column　　B. alter table … modify column…

C. alter table … update …　　D. alter table … update column…

13. 选择当前数据库的命令是________。

A. shows　　B. create　　C. use　　D. show

14. select 语句中删除结果集重复行的命令是________。

A. where　　B. having　　C. distinct　　D. order by

15. 下列选项中属于创建数据库的语句是________。

A. create database　　B. alter database　　C. drop database　　D. 以上都不是

16. 查询 student 表中 id 值在 2 和 7 之间的学生姓名,应该使用关键字________。

A. between and　　B. in　　C. like　　D. OR

17. 用 is null 关键字来判断字段的值是否为空值,那 is null 关键字应该在________之后使用。

A. order by　　B. where　　C. select　　D. limit

18. 在聚合函数中,用来统计记录条数的函数是________。

A. sum()　　B. avg()　　C. max()　　D. count()

19. select 语句中,使用________子句可按指定列的值分组。

A. where　　B. having　　C. group by　　D. order by

20. 要想移除账户,应使用________语句。

A. delete user　　B. drop user　　C. delete priv　　D. drop priv

二、填空题(每空 1 分,共 10 分)

1. ________是在数据库中定义的一些完成特定功能的 SQL 语句的集合。

2. 函数 round(58.3049,2)的返回值是________。

3. bool 型数据用于存储逻辑值,它只有两种状态,即________和________。

4. select 语句中,若希望将查询结果排序,则应在 select 语句中使用 order by 子句,其中,asc 选项表示升序,________选项表示降序。

5. 在 MySQL 中,查询时想要输出显示 2~12 行信息,可用到关键字________。

6. 聚合函数中,用于求出某个字段最大值的函数是________。

7. select 语句的执行过程是从数据库中选取匹配的特定记录和字段,并将这些数据组织成一个结果集,然后以________的形式返回。

8. 在 MySQL 中,清空表的数据可以使用________和________命令实现。

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
add_num: loop
    set @count = @count + 1;
```

```
    if @count = 30
      then leave add_num ;
    endif;
  end loop add_num ;
```

2. select studentno, sname, phone, Email

```
    from student
    where phone regexp '7 $ ';
```

3. create index Multi_Col2

```
on test(first, middle, last);
```

4. delimiter //

```
  create procedure cproc(out par1 int)
  begin
     select  count( * )  into  param1  from  frus;
  end //
  delimiter ;
```

5. mysql -u root -p --vertical

```
 --execute="select * from  pers;" test > D:\pers1.txt
```

6. mysqlbinlog --stop-date="2017-03-30 16:30:30"

```
D:\MySQL\log\binlog\binlog.000027 | mysql - uuser - ppass
```

四、简答题(每题 3 分,共 12 分)

1. 简述数据库系统的组成。
2. 简述利用 truncate table 和 delete 语句删除表数据时有什么区别。
3. 简述并发控制可能产生的原因。
4. 简述在数据库中创建索引的必要性。

五、综合题(每小题 4 分,共 40 分)

MySQL 数据库 teach 中有学生信息表 stu_info 和学生选课成绩表 se_score,创建表结构的命令如下所示,利用 SQL 语言写出以下各题的操作命令。

```
create table stu_info(
  学号 char(4) primary key,
  姓名 char(8) null,
  性别 char(2) null,
  入学成绩 int null,
  出生日期 datetime  null
 );
create table se_score(
  学号 char(4)  not null,
  课程编号 char(5)  not null,
  成绩 int  null,
  primary key (学号,课程编号)
  );
```

1. 为数据库中 se_score 表的“学号”列创建一个 foreign key 约束 fk_se,参照 stu_info 表的“学号”列。
2. 显示平均成绩低于 80 分的学生的学号、姓名。
3. 显示单科成绩最高的学生的学号、姓名及课程编号。

4. 在 stu_info 表中的“学号”列上创建一个普通索引 index_xh。

5. 建立视图 v_score，查询所有学生的姓名和平均分数。

6. 创建一个存储过程 p_delete，删除 se_score 表中指定学号的记录。

7. 在 stu_info 表上创建一个触发器 stu_ trig，当删除 stu_info 表记录时，同步删除 se_score 表中指定的记录。

8. 求出男女学生的平均入学成绩。

9. 为 stu_info 添加一个能够存储照片的字段。

10. 使用 mysqldump 命令备份数据库 teach 中的 stu_info 表和 se_score 表。

MySQL 数据库模拟试题 4 参考答案

一、选择题(每题 1 分,共 20 分)

1～5. CDCCC　6～10. ADAAD　11～15. DBCCA　16～20. ABDCB

二、填空题(每空 1 分,共 10 分)

1. 存储过程
2. 58.30
3. true、false
4. desc
5. limit
6. max()
7. 临时表
8. truncate　delete

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1. 循环执行 count 加 1 的操作。当 count 的值等于 30 时,使用 leave 语句跳出循环。
2. 查询 student 表中学生电话号码尾数为 7 的学生的信息。
3. 使用 create index 建立名称为 MultiCol2 的组合索引。
4. 创建一个获取 frus 表记录条数的存储过程 cproc,利用 count(*)计算后把结果放入参数 par1 中。
5. 使用 MySQL 命令导出 test 数据库中 pers 表中的记录到文本文件,使用--vertical 参数显示结果。
6. 使用 mysqlbinlog 恢复 MySQL 数据库到 2017 年 3 月 30 日 16:30:30 时的状态。

四、简答题(每题 3 分,共 12 分)

1. 数据库系统(Database System,DBS)通常由硬件、软件、数据库和用户组成,管理的对象是数据。其中软件主要包括操作系统、各种宿主语言、实用程序以及数据库管理系统。数据库系统包括四大组件:用户、数据、软件和硬件。

(1) 用户(Users)。用户执行 DDL 语言定义数据库架构,使用 DML 语言新增、删除、更新和查询数据库的数据,通过操作系统访问数据库的数据。按不同角色划分,用户可以分为多种,如终端用户(End-Users)、数据库设计者(Database Designers)、系统分析师(System Analyst,SA)、程序员(Application Programmer)和数据库管理员等。数据库管理员负责创建、监控和维护整个数据库,一般是由业务水平较高、资历较深的人员担任。

(2) 数据(Data)。数据库系统中的数据种类包括永久性数据(Persistent Data)、索引数据(Indexes)、数据字典(Data Dictionary)和事务日志(Transaction Log)等。

(3) 软件(Software)。指在数据库环境中使用的软件,包括数据库管理系统、应用程序和开发工具(Development Tools)等。

(4) 硬件(Hardware)。安装数据库相关软件的硬件设备,包含主机(CPU、内存和网卡

等)、磁盘阵列、光驱和备份装置等。

2. 为什么要用 truncate table 语句代替 delete 语句？当使用 truncate table 语句时，记录的删除是不作记录的。也就是说，这意味着 truncate table 要比 delete 快得多。

3. 多个用户访问同一个数据资源时，可能出现的一些并发问题，并发控制可能产生的原因如下：

(1) 更新丢失。当两个或多个事务选择同一行，每个事务都不知道其他事务的存在。最后的更新将覆盖其他事务所做的更新，从而导致数据丢失。

(2) 不可重复读。当一个事务多次访问同一行且每次读取不同数据时，会出现不可重复读问题。

(3) 幻读。当对某行执行插入或删除操作，而该行属于某事务正在读取的行的范围时，就会出现幻读问题。

(4) 脏读，即读出的是不正确的临时数据。

4. 在数据库中创建索引可以大大提高系统的查询性能，加快数据的检索速度，可以保证每一行数据的唯一性，加速表与表之间的连接。

同时，在使用 order by 和 group by 子句进行数据检索时，可以显著减少查询中分组和排序的时间。在查询过程中，使用查询优化器，提高系统性能。

五、综合题(每小题 4 分，共 40 分)

1.
```
alter table se_score
add constraint fk_se foreign key (学号)
references student(学号)
on update cascade
on delete cascade;
```

2.
```
select stu_info.学号,stu_info.姓名 from stu_info
where stu_info.学号 in (
    select se_score.学号
    from se_score
    group by se_score.学号
    having avg(se_score.成绩)< 80);
```

3.
```
select  stu_info.学号,stu_info.姓名,stu_info.课程编号
from  stu_info,se_score
where stu_info.学号 = (select se_score.学号
                from se_score
                where se_score.成绩 =
               (select max(se_score.成绩)  from  se_score));
```

4.
```
create index Index_xh on stu_info(学号);
```

5.
```
create view v_score
as
select stu_info.姓名,avg(成绩)as '平均成绩'
from stu_info,se_score
where stu_info.学号 = se_score.学号
group by stu_info.姓名;
```

6.
```
delimiter //
create  procedure p_delete(in xh char(4))
begin
```

```
  delete  from  se_score where 学号 =  xh;
End //
delimiter ;
```

7.
```
delimiter//
create  trigger  stu_trig
after  delete
on  stu_info  for each row
begin
        select @sno = 学号 from old;
        delete from se_score Where 学号  = @sno ;
      end//
delimiter ;
```

8.
```
select avg(stu_info.入学成绩) as 平均入学成绩 ,性别
from stu_info
group by 性别
```

9.
```
alter table stu_info
add fields mediumblob comment'照片';
```

10.
```
mysqldump －u root －p teaching stu_info  se_score
> d:/bak/teach_se.sql
Enter password: ******
```

MySQL 数据库模拟试题 5

说明：本试题为笔试方式，考试时间 100 分钟，总计 100 分。

一、选择题（每题 1 分，共 20 分）

1. 下列________语句用于创建视图。

   A. create triger　　　　B. create procedure

   C. create view　　　　D. create table

2. 在关系数据库中，为了简化用户的查询操作，而又不增加数据的存储空间，则应该创建的数据库对象是________。

   A. table　　B. index　　C. cursor　　D. view

3. 在 MySQL 数据库自带的命令工具中，________工具执行管理操作的客户程序。例如创建或删除数据库，或者清除所有缓存的主机和所有日志等。

   A. mysql　　　　B. mysqladmin

   C. mysqlshow　　　　D. mysqlbinlog

4. 关于 truncate table 描述不正确的是________。

   A. truncate 将删除表中的所有数据

   B. 表中包含 auto_incerement 列，使用 truncate table 可以重置序列值为该列的初始值

   C. truncate 操作比 delete 操作占用资源多

   D. truncate table 删除表，然后重新构建表

5. 下列几种情况下，不适合创建索引的是________。

   A. 列的取值范围很小　　　　B. 用作查询条件的列

   C. 频繁搜索范围的列　　　　D. 连接中频繁使用的列

6. 在视图上不能完成的操作是________。

   A. 更新基本表数据　　　　B. 查询

   C. 在视图上定义新的基本表　　　　D. 在视图上定义新视图

7. 创建索引是为了________。

   A. 提高存取速度　　　　B. 减少 I/O

   C. 节约空间　　　　D. 减少缓冲区个数

8. 为了使索引键的值在基本表中唯一，在创建索引的语句中应使用保留字________。

   A. unique　　B. count　　C. distinct　　D. union

9. 关于视图，下列哪一个说法是错误的？________

   A. 视图是一种虚拟表　　　　B. 视图中也保存有数据

   C. 视图也可由视图派生出来　　　　D. 视图保存在 select 查询中

10. 在创建数据库时，每个数据库都对应存放在一个与数据库同名的________中。

    A. 文件　　B. 文件夹　　C. 路径　　D. 以上都不是

11. MySQL 使用________文件中的配置参数。

A. my-larger.ini　B. my-small.ini　C. my-huge.ini　D. my.ini

12. 下列选项中属于删除数据库的语句是________。

A. create database　B. use database

C. drop database　D. 以上都不是

13. 下列聚合函数不忽略空值(null)的是________。

A. sum(列名)　B. max(列名)　C. count(*)　D. avg(列名)

14. 下列________语句用于释放预处理语句。

A. prepare　B. execute　C. deallocate　D. using

15. 在 SQL 中,下列涉及空值的操作,不正确的是________。

A. age is null　B. age is not null

C. age=null　D. not(age is null)

16. select 语句中,________子句用于选择满足给定条件的元组。

A. where　B. having　C. group by　D. order by

17. 为了提高特定查询的速度,对 SC(Sno, Cno, degree)关系创建唯一性索引,应该创建在________个(组)属性上。

A. (Sno, Cno)　B. (Sno, degree)　C. (Cno, degree)　D. degree

18. 下列函数中,能返回数值表达式整数部分值的是________。

A. abs()　B. int()　C. sign()　D. sqrt()

19. 在 SQL 语句中,检索去掉重复数据,则在 select 中使用关键字________。

A. all　B. union　C. like　D. distinct

20. 下列函数中,能获取当前系统日期和时间的函数是________。

A. curdate()　B. curtime()　C. sysdate()　D. date_format()

二、填空题(每空 1 分,共 10 分)

1. 数据模型通常由 3 部分组成,即________、数据操作和完整性约束。

2. 以@@作为首部的变量在 MySQL 系统中称为________。

3. 如果使用________关键字,则即使在更新过程中出现错误,更新语句也不会中断。

4. 在 where 子句中,关系运算符“<>”与关系运算符________等价,而逻辑运算符“&&”则与________等价。

5. 要查询 student 表中 name 字段值以字符“m”开始,以字符“d”结束的记录应该在 where 子句后跟 like ________。

6. 聚合函数中,用于求出某个字段平均值的函数是________。

7. 使用 order by 对查询结果进行排序时,默认是按________顺序排列,也可以按________顺序排列。

8. ________是特殊类型的存储过程,它能在任何试图改变表中由触发器保护的数据时执行。

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
select  case  2  when 1  then 'one'
                 when 2  then 'two'
```

11. MySQL 使用________文件中的配置参数。

A. my-larger. ini　B. my-small. ini　C. my-huge. ini　D. my. ini

12. 下列选项中属于删除数据库的语句是________。

A. create database　B. use database

C. drop database　D. 以上都不是

13. 下列聚合函数不忽略空值(null)的是________。

A. sum(列名)　B. max(列名)　C. count(*)　D. avg(列名)

14. 下列________语句用于释放预处理语句。

A. prepare　B. execute　C. deallocate　D. using

15. 在 SQL 中,下列涉及空值的操作,不正确的是________。

A. age is null　B. age is not null

C. age=null　D. not(age is null)

16. select 语句中,________子句用于选择满足给定条件的元组。

A. where　B. having　C. group by　D. order by

17. 为了提高特定查询的速度,对 SC(Sno, Cno, degree)关系创建唯一性索引,应该创建在________个(组)属性上。

A. (Sno, Cno)　B. (Sno, degree)　C. (Cno, degree)　D. degree

18. 下列函数中,能返回数值表达式整数部分值的是________。

A. abs()　B. int()　C. sign()　D. sqrt()

19. 在 SQL 语句中,检索去掉重复数据,则在 select 中使用关键字________。

A. all　B. union　C. like　D. distinct

20. 下列函数中,能获取当前系统日期和时间的函数是________。

A. curdate()　B. curtime()　C. sysdate()　D. date_format()

二、填空题(每空 1 分,共 10 分)

1. 数据模型通常由 3 部分组成,即________、数据操作和完整性约束。

2. 以@@作为首部的变量在 MySQL 系统中称为________。

3. 如果使用________关键字,则即使在更新过程中出现错误,更新语句也不会中断。

4. 在 where 子句中,关系运算符“<>”与关系运算符________等价,而逻辑运算符“&&”则与________等价。

5. 要查询 student 表中 name 字段值以字符“m”开始,以字符“d”结束的记录应该在 where 子句后跟 like ________。

6. 聚合函数中,用于求出某个字段平均值的函数是________。

7. 使用 order by 对查询结果进行排序时,默认是按________顺序排列,也可以按________顺序排列。

8. ________是特殊类型的存储过程,它能在任何试图改变表中由触发器保护的数据时执行。

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
select  case  2  when 1  then 'one'
                 when 2  then 'two'
```

11. MySQL 使用________文件中的配置参数。

A. my-larger. ini　　B. my-small. ini　　C. my-huge. ini　　D. my. ini

12. 下列选项中属于删除数据库的语句是________。

A. create database　　B. use database

C. drop database　　D. 以上都不是

13. 下列聚合函数不忽略空值(null)的是________。

A. sum(列名)　　B. max(列名)　　C. count(*)　　D. avg(列名)

14. 下列________语句用于释放预处理语句。

A. prepare　　B. execute　　C. deallocate　　D. using

15. 在 SQL 中,下列涉及空值的操作,不正确的是________。

A. age is null　　B. age is not null

C. age=null　　D. not(age is null)

16. select 语句中,________子句用于选择满足给定条件的元组。

A. where　　B. having　　C. group by　　D. order by

17. 为了提高特定查询的速度,对 SC(Sno, Cno, degree)关系创建唯一性索引,应该创建在________个(组)属性上。

A. (Sno, Cno)　　B. (Sno, degree)　　C. (Cno, degree)　　D. degree

18. 下列函数中,能返回数值表达式整数部分值的是________。

A. abs()　　B. int()　　C. sign()　　D. sqrt()

19. 在 SQL 语句中,检索去掉重复数据,则在 select 中使用关键字________。

A. all　　B. union　　C. like　　D. distinct

20. 下列函数中,能获取当前系统日期和时间的函数是________。

A. curdate()　　B. curtime()　　C. sysdate()　　D. date_format()

二、填空题(每空 1 分,共 10 分)

1. 数据模型通常由 3 部分组成,即________、数据操作和完整性约束。

2. 以@@作为首部的变量在 MySQL 系统中称为________。

3. 如果使用________关键字,则即使在更新过程中出现错误,更新语句也不会中断。

4. 在 where 子句中,关系运算符“<>”与关系运算符________等价,而逻辑运算符“&&”则与________等价。

5. 要查询 student 表中 name 字段值以字符“m”开始,以字符“d”结束的记录应该在 where 子句后跟 like ________。

6. 聚合函数中,用于求出某个字段平均值的函数是________。

7. 使用 order by 对查询结果进行排序时,默认是按________顺序排列,也可以按________顺序排列。

8. ________是特殊类型的存储过程,它能在任何试图改变表中由触发器保护的数据时执行。

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
select   case  2  when 1  then 'one'
                  when 2  then 'two'
```

11. MySQL 使用________文件中的配置参数。

A. my-larger. ini　B. my-small. ini　C. my-huge. ini　D. my. ini

12. 下列选项中属于删除数据库的语句是________。

A. create database　B. use database

C. drop database　D. 以上都不是

13. 下列聚合函数不忽略空值(null)的是________。

A. sum(列名)　B. max(列名)　C. count(*)　D. avg(列名)

14. 下列________语句用于释放预处理语句。

A. prepare　B. execute　C. deallocate　D. using

15. 在 SQL 中,下列涉及空值的操作,不正确的是________。

A. age is null　B. age is not null

C. age=null　D. not(age is null)

16. select 语句中,________子句用于选择满足给定条件的元组。

A. where　B. having　C. group by　D. order by

17. 为了提高特定查询的速度,对 SC(Sno, Cno, degree)关系创建唯一性索引,应该创建在________个(组)属性上。

A. (Sno, Cno)　B. (Sno, degree)　C. (Cno, degree)　D. degree

18. 下列函数中,能返回数值表达式整数部分值的是________。

A. abs()　B. int()　C. sign()　D. sqrt()

19. 在 SQL 语句中,检索去掉重复数据,则在 select 中使用关键字________。

A. all　B. union　C. like　D. distinct

20. 下列函数中,能获取当前系统日期和时间的函数是________。

A. curdate()　B. curtime()　C. sysdate()　D. date_format()

二、填空题(每空 1 分,共 10 分)

1. 数据模型通常由 3 部分组成,即________、数据操作和完整性约束。

2. 以@@作为首部的变量在 MySQL 系统中称为________。

3. 如果使用________关键字,则即使在更新过程中出现错误,更新语句也不会中断。

4. 在 where 子句中,关系运算符“<>”与关系运算符________等价,而逻辑运算符“&&”则与________等价。

5. 要查询 student 表中 name 字段值以字符“m”开始,以字符“d”结束的记录应该在 where 子句后跟 like ________。

6. 聚合函数中,用于求出某个字段平均值的函数是________。

7. 使用 order by 对查询结果进行排序时,默认是按________顺序排列,也可以按________顺序排列。

8. ________是特殊类型的存储过程,它能在任何试图改变表中由触发器保护的数据时执行。

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
select  case  2  when 1  then 'one'
                 when 2  then 'two'
```

```
                    else 'more'
            end case ;
```

2.
```
alter table book
add index BkNameIdx( bookname(30) );
show index from  book \G
```

3.
```
select sno,sname  from   student
where  exists(select * from  score
                      where students.sno = score.sno   and final > 90);
```

4.
```
alter table student
add column comment text not null;
```

5.
```
mysqldump  - u root  - p booksDB books > D:/backup/books_20170707.sql
```

6.
```
C:\> mysqlbinlog  D:/MySQL/log/binlog.000011
```

四、简答题(每题 3 分,共 12 分)

1. 简要说明事务的隔离性的特点。
2. 使用直接复制方法实现数据库备份与恢复时,需要注意哪些事项?
3. 简述数据完整性的实现方法和作用。
4. 简述存储过程的优点。

五、综合题(每小题 4 分,共 40 分)

使用 MySQL 语句完成操作,在数据库 product_sales 中参考下列表的结构:

```
product(产品编号 char(10) primary key,
  分类号 char(6),
  产品名 char(30) not null unique,
  库存量 int,
  订货点 int,
  单价 float(5,2))
order_details(产品编号 char(10) references product(产品编号),
  订单编号 char(10) references  c_order(订单编号),
  订购数量(int))
c_order(订单编号 char(10) primary key,
  客户编号 char(10) references consumer(客户编号),
  订购日期 datetime)
consumer(客户编号 char(10)  primary key,
  客户名称 varchar(30),
  联系人 char(16),
  联系电话 char(12))
```

1. 查询单价介于 50 元和 100 元的产品。
2. 查询产品订购总数量大于 3500 的产品名称及产品订购总数量。
3. 查询 2017 年 10 月开始没有下订单的客户信息。
4. 将订单编号开头为“2017”的订单所订购的产品数量加倍。
5. 删除 c_order 表中所有‘2015-10-01’以前的订单信息。
6. 查询所有订单编号以“2017”开头的产品的订购客户信息。
7. 给 consumer 表增加一个列“备注”,数据类型为 text,不允许为空。
8. 按产品编号分组,计算 product 表中的每种产品的库存量及资金额度。输出资金额

度较大的前10种产品数据。

9．使用alter table语句在order_details表上的订单编号列添加索引，取名为index_details。

10．设置触发器tri_update，实现在product表中修改产品编号时，可自动修改课程在order_details上的产品编号。

MySQL 数据库模拟试题 5 参考答案

一、选择题(每题 1 分,共 20 分)

1～5. CDACA　6～10. CAABB　11～15. DCCCC　16～20. AABDC

二、填空题(每空 1 分,共 10 分)

1. 数据结构
2. 全局变量
3. ignore
4. !=　and
5. 'm%d'
6. avg()
7. 升序　降序
8. 触发器

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1. 使用 case...when 语句执行分支操作。
2. 使用 alter table 在 bookname 字段上添加索引,使用 show index 语句查看表中的索引定义。
3. 查询有科目成绩超过 90 分的学生的学号和姓名。
4. 给 student 表增加一个列"备注",数据类型为 text,不允许为空。
5. 备份 booksDB 数据库中的 books 表。
6. 使用 mysqlbinlog 查看二进制日志。

四、简答题(每题 3 分,共 12 分)

1. 隔离性是指每个事务在其自己的会话空间发生,和其他发生在系统中的事务隔离,而且事务的结果只有在完全被执行后才能看到。即一个事务内部的操作及使用的数据对并发的其他事务是隔离的,并发执行的各个事务之间不能互相干扰。该机制是通过对事务的数据访问对象加适当的锁,排斥其他事务对同一数据库对象的并发操作来实现的。

2. 使用直接从一个 MySQL 服务器复制文件到另一个服务器的方法,需要特别注意以下两点:

(1) 两个服务器必须使用相同或兼容的 MySQL 版本。

(2) 两个服务器必须硬件结构相同或相似,除非要复制的表使用 MyISAM 存储格式,这是因为这种表可以为在不同的硬件体系中共享数据提供保证。

3. 在定义和修改表结构时,可以通过定义与该表相关的完整性约束条件,实现数据完整性约束。这些约束条件主要包括 not null(非空约束)、primary key(主键约束)、unique(唯一性约束)、foreign key(外键参照完整性约束)以及 check(检查约束)。

数据完整性约束条件都被存入系统的数据字典中,当用户操作表中的数据时,由数据库管理系统自动检查该操作是否违背这些完整性约束条件。如果完整性约束条件涉及该表的

多个属性列，则必须定义在表级上，其他情况则既可以定义在列级上也可以定义在表级上。通过数据约束条件的实际应用，可以实现监视数据输入过程中出现重复或错误等，保证数据的正确性、精确性。

4. 存储过程是 MySQL 语句的集合。存储过程可以使得程序执行效率更高、安全性更好，因为过程建立之后已经编译并且存储到数据库，直接写 SQL 就需要先分析再执行，因此过程效率更高，直接写 SQL 语句会带来安全性问题。建立过程不会很耗系统资源，因为过程只是在调用才执行。

五、综合题（每小题 4 分，共 40 分）

1.
```
select * from product
where 单价 between 50 and 100;
```

2.
```
select p.产品名称, sum(od.订购数量)
from product as p, Order_details as od
where p.产品编号 = od.产品编号
group by od.产品编号,p.产品名称
having sum(od.订购数量)> 3500;
```

3.
```
select  *  from consumer
where 客户编号 not  in
              (select  distinct 客户编号
               from   c_order
               where  订购日期> = 2017 - 10 - 01);
```

4.
```
update Order_details set 订购数量 = 订购数量 * 2
where 订单编号 like '2017 % ';
```

5.
```
delete c_order
where 订购日期<'2015 - 10 - 01'
```

6.
```
select * from consumer, c_order
on consumer.客户编号 = c_order.客户编号
where c_order.订单编号 like '2017 % ';
```

7.
```
alter table consumer
add column 备注 text not null;
```

8.
```
select  产品编号,count(库存量),count(库存量) * 单价
from  product
group by 产品编号
order  by  count(库存量) *  单价 desc
limit 10;
```

9.
```
alter table order_details
add index index_details (订单编号);
```

10.
```
create trigger tri_update
after update on product for each row
update order_details set order_details.产品编号 = new.产品编号
where order_details.产品编号 = old.产品编号;
```

MySQL 数据库模拟试题 6

说明：本试题为笔试方式，考试时间 100 分钟，总计 100 分。

一、选择题（每题 1 分，共 20 分）

1. 运算符%能够用于________类型的数据上。

   A. float　　B. char　　C. smallint　　D. date

2. 执行下面语句的结果是________。

```
Select  upper(ltrim(rtrim('  Hello,world  ')));
```

   A. 'HELLO，WORLD'　　B. 'Hello，world　'

   C. 'Hello，world'　　D. '　HELLO，WORLD　'

3. 在关系数据库中，能够唯一地标识一个记录的属性或属性的组合，称为________。

   A. 主码　　B. 属性　　C. 关系　　D. 域

4. 在 SQL select 语句中用于实现关系的选择运算的短语是________。

   A. for　　B. while　　C. where　　D. condition

5. 使用 show 语句查看现在的数据库中表的命令是________。

   A. show databases；　　B. create database；

   C. use mysqltest；（回车）　　D. show tables；

6. 用于显示表的结构的命令是________。

   A. create table　　B. describe

   C. insert　　D. update

7. 设有图书管理数据库：

```
图书(总编号 C(6),分类号 C(8),书名 C(16),作者 C(6),出版单位 C(20),单价 N(6,2))
读者(借书证号 C(4),单位 C(8),姓名 C(6),性别 C(2),职称 C(6),地址 C(20))
借阅(借书证号 C(4),总编号 C(6),借书日期 D(8))
```

对于图书管理数据库，检索借阅了《网络技术》一书的借书证号。下面 SQL 语句正确的是________。

```
select 借书证号 from 借阅 where 总编号 = ________;
```

   A.（select 借书证号 from 图书 where 书名＝"网络技术"）

   B.（select 总编号 from 图书 where 书名＝"网络技术"）

   C.（select 借书证号 from 借阅 where 书名＝"网络技术"）

   D.（select 总编号 from 借阅 where 书名＝"网络技术"）

8. 修改自己的 MySQL 服务器密码的命令是________。

   A. mysql　　B. grant

   C. set password　　D. change password

9. 关于 select 语句以下哪一个描述是错误的？________

A. select 语句用于查询一个表或多个表的数据

B. select 语句属于数据操作语言(DML)

C. select 语句的列必须是基于表的列的

D. select 语句表示数据库中一组特定的数据记录

10. 改变表名的指令是________。

A. rename table t1 rename to t2；　　B. rename table t1 to t2；

C. update name t1 to t2；　　D. change name t1 to t2；

11. 关系数据库中，外码(foreign key)是________。

A. 在一个关系中定义了约束的一个或一组属性

B. 在一个关系中定义了缺省值的一个或一组属性

C. 在一个关系中的一个或一组属性是另一个关系的主码

D. 在一个关系中用于唯一标识元组的一个或一组属性

12. 在 MySQL 中，通常使用________语句来指定一个已有数据库作为当前工作数据库。

A. using　　B. used　　C. uses　　D. use

13. MySQL 中存储用户全局权限的表是________。

A. table_priv　　B. procs_priv　　C. columns_priv　　D. user

14. 实现数据恢复的命令是________。

A. mysqldump　　B. mysql　　C. backup　　D. return

15. 给名字是 liwen 的用户分配对数据库 studb 中的 stuinfo 表的查询和插入数据权限的语句是________。

A. grant select，insert on studb. stuinfo for'liwen'@'localhost'

B. grant select，insert on studb. stuinfo to 'liwen'@'localhost'

C. grant'liwen'@'localhost' to select，insert for studb. stuinfo

D. grant 'liwen'@'localhost' to studb. stuinfo on select，insert

16. 以下________语句用于撤销权限。

A. delete　　B. drop　　C. revoke　　D. update

17. 下面选项中不属于存储过程的优点的是________。

A. 增强代码的重用性和共享性　　B. 可以运行加快速度，减少网络流量

C. 可以作为安全性机制　　D. 编辑简单

18. “一个事务对数据库的所有操作，是一个不可分割的工作单元，这些操作要么全部执行要么什么也不做”，这是事务的________。

A. 原子性　　B. 一致性　　C. 隔离性　　D. 持久性

19. 若事务 T 对数据 R 已经加了 X 锁，则其他事务对数据 R ________。

A. 可以加 S 锁不能加 X 锁　　B. 不能加 S 锁可以加 X 锁

C. 可以加 S 锁也可以加 X 锁　　D. 不能加任何锁

20. 并发操作会带来________数据不一致性。

A. 丢失更新、不可重复读、脏读、死锁　　B. 不可重复读、脏读、死锁

C. 丢失更新、不可重复读、脏读　　D. 丢失更新、不可重复读、死锁

二、判断题(每题 1 分,共 10 分)

1. or 关键字和 and 关键字可以一起使用,但是 or 的优先级高于 and。（　　）
2. 在关键字 in 之前使用 not 关键字可以查询不在指定集合范围内的记录。（　　）
3. distinct 关键字只能作用于一个字段。（　　）
4. 在对字符串进行模糊查询中,一个下画线通配符可匹配多个字符。（　　）
5. having 关键字和 where 关键字后可以使用聚合函数。（　　）
6. 触发器与约束发生冲突,触发器将不执行。（　　）
7. 创建唯一性索引的列可以有一些重复的值。（　　）
8. 恢复是利用冗余数据来重建数据库的。（　　）
9. 事务是并发控制的基本单位。（　　）
10. 视图本身没有保存数据,而是保存一条查询语句。（　　）

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1.
```
Create trigger trigger_update after update on kc for each row
update xs_kc set xs_kc.kno = new.kno where xs_kc.kno = old.kno;
```

2.
```
select score.sno,score.cno,tname
from   teacher,teach, score
where  teach.cno =  score.cno  and teacher.tno = teach.tno  and  tname = '陈宜';
```

3.
```
create view stu_glass (id,name, glass)
as
select student.s_id,student.name ,stu_info.glass
from  student ,stu_info
where student.s_id = stu_info.s_id;
select * from  stu_glass;
```

4.
```
delimiter $$
create  procedure  select_score(in xh char(6),kch  char(6))
begin
   select * from score where s_no = xh  and c_no = kch;
end $$
delimiter ;
```

5.
```
insert into user (Host,User,Password)
values('localhost','customer', password('custword'));
```

6.
```
mysql -u root -p booksDB < D:/backup/booksdb_201707.sql
```

四、简答题(每题 3 分,共 12 分)

1. 简述利用 union 运算符合并结果集时,对结果集有什么要求。
2. 简述使用 PHP 进行 MySQL 数据库编程的基本步骤。
3. 简述游标在存储过程中的作用。
4. 简述使用索引的优点。

五、按要求绘制 E-R 图操作,并设计关系(第 1 题 8 分,第 2、3 每题 4 分,共 16 分)

某企业集团有若干工厂,每个工厂生产多种产品,且每一种产品可以在多个工厂生产,每个工厂按照固定的计划数量生产产品;每个工厂聘用多名职工,且每名职工只能在一个工厂工作,工厂聘用职工有聘期和工资。其中:

工厂的属性有工厂编号、厂名、地址；

产品的属性有产品号、产品名、规格；

职工的属性有职工号、姓名、电话、地址。

1. 设计满足上述要求的 E-R 图，注明属性。（根据自己的理解，用下画线标明每个实体中的主码）。

2. 将 E-R 图转换为等价的关系模型方式。

3. 指出转换结果中每个关系模式的主码和外码。

六、根据下列数据库中表的结构，写出 MySQL 语句（每小题 3 分，共 24 分）

```
student(
 学号 char(8) primary key,
 姓名 char(8),
 班级 char(10),
 性别 char(2),
 出生日期 datetime,
 出生城市 char(20),
 入学成绩 tinyint)
course(
 课程号 char(6) primary key,
 课程名 char(20))
c_select(
 学期 char(2),
 学号 char(8) references student(学号),
 课程号 char(6) references course(课程号),
 成绩 tinyint check(成绩> = 0 and 成绩< = 100)
```

1. 查询缺少成绩的 student 的学号和相应的课程号。

2. 查询 1803 物流班全体学生的学号与姓名，且按照入学成绩的降序排列。

3. 统计班级的平均入学总分在 650 分以上的班级和这些班级的平均入学总分。

4. 查询选修了‘实用英语’课程的学生的学号，以及实用英语的成绩。

5. 查询第 2 学期所选课程平均成绩前 3 名的那些学生的学号和姓名。

6. 在 student 表上增加一个 image 列，存放照片。

7. 修改 student 表上“出生日期”列，使其类型为 date。

8. 将 course 表的 select 和 update 权限授予数据库用户 user2。

MySQL 数据库模拟试题 6 参考答案

一、选择题(每题 1 分,共 20 分)

1～5. CAACD　6～10. BBCCB　11～15. CDDBB　16～20. CDADC

二、判断题(每题 1 分,共 10 分)

1. F　2. T　3. F　4. F　5. F　6. T　7. F　8. T　9. T　10. T

三、阅读程序,指出下列程序段的功能(每题 3 分,共 18 分)

1. 假设之前创建的 KC 表没有设置外键级联策略,设置触发器,实现在 KC 表中修改课程 ID 号时,可自动修改课程在 xs_kc 上的课程 ID 号。

2. 查询教师“陈宜”授课的课程号和课程名。

3. 在表 student 和表 stu_info 上创建视图 stu_glass,并浏览视图。

4. 创建存储过程 select_score,输出指定课程号和学号的学生成绩。

5. 使用 insert 创建一个用户名为 customer 的新账户,主机名称为 localhost,密码为 custword。

6. 使用 MySQL 命令将 D:\backup\booksdb_201707.sql 文件中的备份导入到数据库中。

四、简答题(每题 3 分,共 12 分)

1. union 运算符可以将两个或两个以上的 select 语句的查询结果集合合并成一个结果集合显示。合并结果集时,查询结果的列标题为第一个查询语句的列标题。因此,要定义列标题必须在第一个查询语句中定义。要对联合查询结果排序时,也必须使用第一查询语句中的列名、列标题或者列序号。

在使用 union 运算符时,应保证每个查询语句的选择列表中有相同数量的表达式,并且每个查询选择表达式应具有相同的数据类型,或是可以自动将它们转换为相同的数据类型。在自动转换时,对于数值类型,系统将低精度的数据类型转换为高精度的数据类型。

2. 使用 PHP 进行 MySQL 数据库编程的基本步骤如下:

(1) 首先建立与 MySQL 数据库服务器的连接。

(2) 然后选择要对其进行操作的数据库。

(3) 再执行相应的数据库操作,包括对数据的添加、删除、修改和查询等。

3. 游标是一个被 select 语句检索出来的结果集。在存储了游标后,应用程序或用户就可以根据需要滚动或浏览其中的数据。

4. 在数据库中使用索引的优点如下。

(1) 加速数据检索:索引能够以一列或多列值为基础实现快速查找数据行。

(2) 优化查询:查询优化器是依赖于索引起作用的,索引能够加速连接、排序和分组等操作。

(3) 强制实施行的唯一性:通过给列创建唯一索引,可以保证表中的数据不重复。

需要注意的是，索引并不是越多越好，要正确认识索引的重要性和设计原则，创建合适的索引。

五、按要求绘制 E-R 图操作，并设计关系（第 1 题 8 分，第 2、3 每题 4 分，共 16 分）

1. 绘制 E-R 图如图 1 所示。

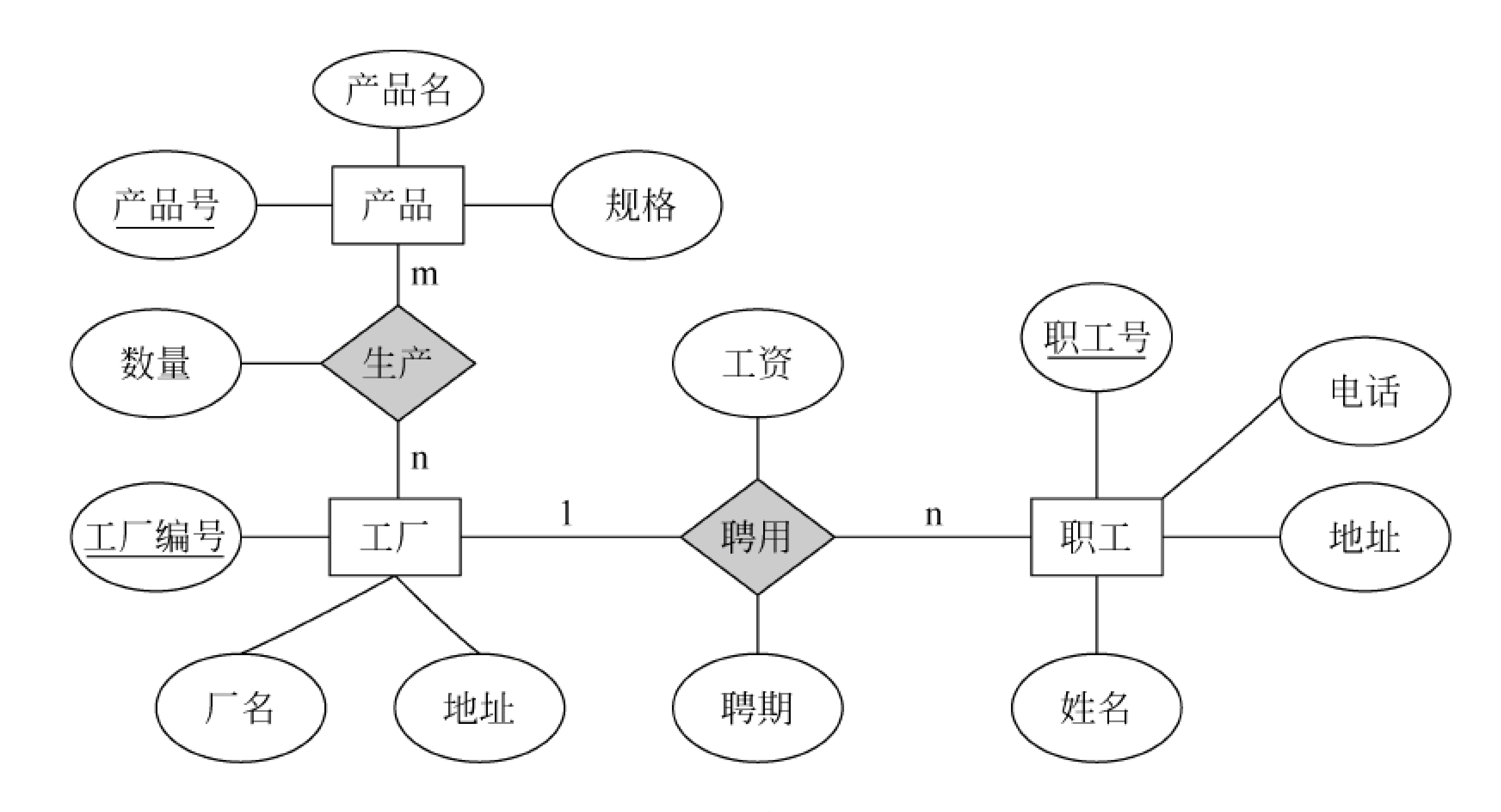


图 1　企业产品 E-R 图

2. 将 E-R 图转换为等价的关系模型方式。

工厂(工厂编号,厂名,地址)
产品(产品号,产品名,规格)
职工(职工号,姓名,地址,电话,工厂编号,聘期,工资)
生产(工厂编号,产品编号,数量)

3. 指出转换结果中每个关系模式的主码和外码。

工厂关系	主码：工厂编号，外码：无
产品关系	主码：产品编号，外码：无
职工关系	主码：职工号，外码：工厂编号
生产关系	主码：工厂编号和产品编号，外码：工厂编号和产品编号

六、根据下列数据库中表的结构，写出 MySQL 语句（每小题 3 分，共 24 分）

1.
```
select 学号,课程号  from c_select
where 成绩 is null;
```

2.
```
select 学号,姓名  from student
where 班级 = '1803 物流'
order by 入学成绩 desc;
```

3.
```
select  班级,avg(入学成绩)  from  student
group by  班级;
having avg(入学成绩)> 650;
```

4.
```
select 学号,成绩  from  c_select
where  课程号 = (select 课程号
                 from  course
                 where  课程名 = '实用英语'
                 );
```

5.
```
select c_select.学号,student .姓名
from c_select,student
where c_select.学期 = '2'and c_select.学号 = student .学号
group byc_select.学号
order by avg(成绩)
limit 3;
```

6. alter table student add column image blob;

7. alter table student modify column 出生日期 date;

8. grant select, update on course touser2@localhost;

图 书 资 源 支 持

感谢您一直以来对清华版图书的支持和爱护。为了配合本书的使用，本书提供配套的资源，有需求的读者请扫描下方的“书圈”微信公众号二维码，在图书专区下载，也可以拨打电话或发送电子邮件咨询。

如果您在使用本书的过程中遇到了什么问题，或者有相关图书出版计划，也请您发邮件告诉我们，以便我们更好地为您服务。

我们的联系方式：

地　　址：北京海淀区双清路学研大厦 A 座 707

邮　　编：100084

电　　话：010－62770175－4604

资源下载：http://www.tup.com.cn

电子邮件：weijj@tup.tsinghua.edu.cn

QQ：883604（请写明您的单位和姓名）



书圈

用微信扫一扫右边的二维码，即可关注清华大学出版社公众号“书圈”。